FLOWLOGIC 6

BEGINNERS GUIDE

myFLOWLAB

transform learning to innovation

BASE UNIT

Beginners Guide

Prototyping and Programming using FLOWLOGIC 6

REV 1.1

In support of

GLOBAL STEM INITIATIVE

About myFlowlab Beginners Guide

The goal of myFlowlab Beginners Guide is to introduce beginners to the world of interactive ICT, Electronic, Sensors, Motors and Programming Algorithm through a prototyping and playful experiments innovating real-world solutions using FlowLogic 6 - a Flowchart based programming language and Arduino Uno board . FlowLogic 6 enables interactive control and real-time monitoring of devices connected to the board. You are able to explore from LED to Robotics, IOT applications and beyond. The limitation is your creativity.

We made it easy and fast for beginners to get started.

This guide offer details from safety, preparation to step by step guides to experiment components included in the starter kit to build and investigate real world solutions.

Once you have mastered the contents of this manual you will be ready to embark into more complex activities using advance electronic components and Algorithm using FlowLogic 6 or Text based programming like C/C++, python..

About myFlowLab

myFlowLab is fully owned by Matroll Solutions, a company founded by a team of engineers and is an energetic young company seeking to make Innovation fun, accessible, and approachable to everyone - from kids in elementary school to University students.

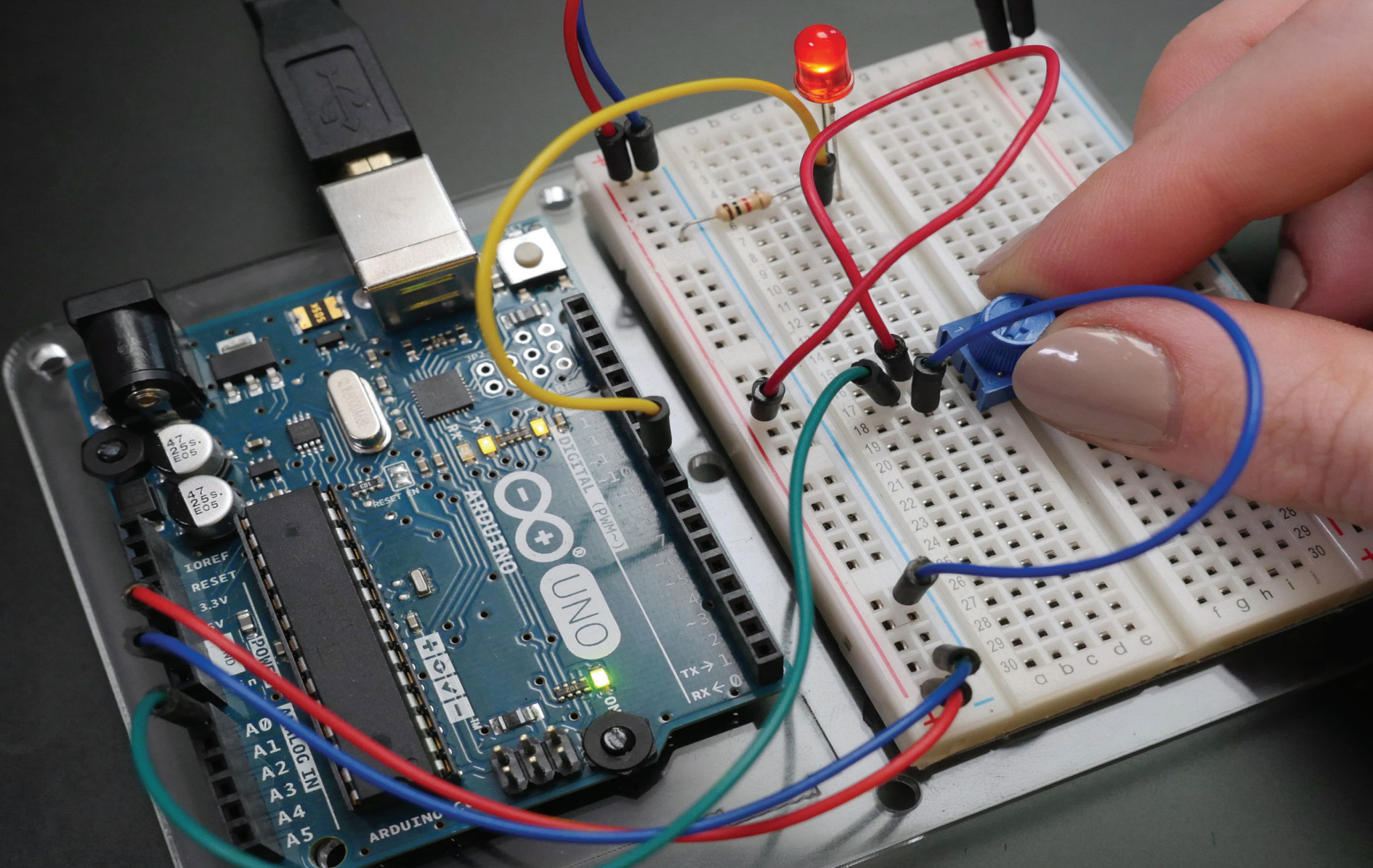
About Problem

We strive to deliver the highest level of quality in each and every thing we produce. If you ever find a confusing instruction, a missing piece, or would just like to ask a question, we'll try our best to help out.
Email us at: help@myflowlab.com (we like hearing about problems it helps us improve future versions)

CONTENT

“You have to learn the rules of the game. And then you have to play better than anyone else.”

Albert Einstein

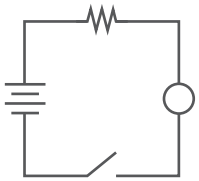

UNO
ARDUINO
DIGITAL (PWM~)
ANALOG IN
IOREF
RESET
3.3V
A0
A1
A2
A3
A4
A5
TX→
RX←

CHAPTER 1

Getting Started

An Overview

In this Guide we will explore step by step various experiments using components included in our myFLOWLAB™ Inventor kit. You are encouraged to carry out all the experiments examples shown in this chapter as this is the foundation towards greatest innovation and real world project development.

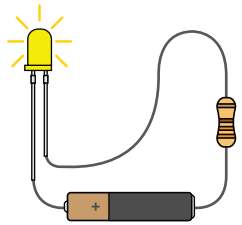


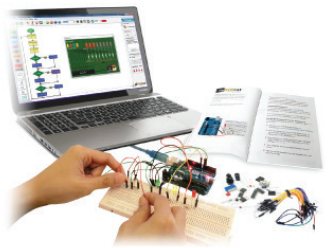
WHAT IS AN ELECTRICAL CIRCUIT?

A circuit is basically an electronics loop with a starting point and an ending point - with any number of components in between. Circuits can include resistors, diodes, and inductors, sensors of all sizes and shapes, motors, and any other handful of hundreds of thousands of components. Circuits are usually divided into three categories – analog circuits, digital circuits, or mixed-signal circuits. In this guide, you will explore all three sets of circuits.

THE WORLD RUNS ON CIRCUITS

Everywhere you look, you'll find circuits. The cell phone in your pocket, the computer that controls your car's emissions system, your video game console - all these things are chock full of circuits. In this guide, you'll experiment with some simple circuits and learn the gist of the world of embedded electronics.





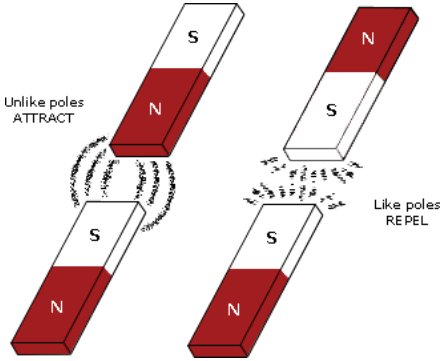
SIMPLE AND COMPLEX CIRCUITS

In this guide, you will be primarily exploring experiments using simple circuits - but that doesn't mean you can't do amazing things. When you've finished our myFLOWLAB™ Playground Starter Kit, your knowledge of circuits will enable you to explore amazing projects and unleash the power of your imagination.

CHAPTER 2

Easy Electronic

We explain some of the basic principles behind working with electronics in simple terms to help you get started with your own practical projects.

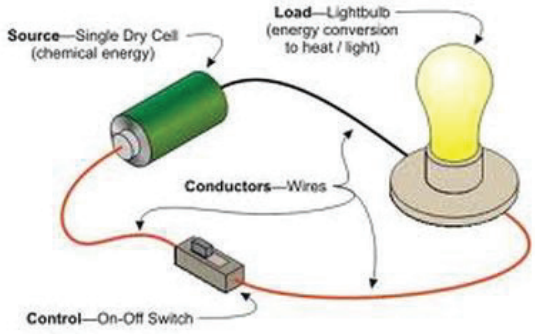


Playing with mySTEM Exploration kit is going to involve using two main disciplines; developing program with FlowLogic 6 and working with electronics. Don't worry if you are not familiar with either, as we will explain most of the important information as we work through our experiments. But before we start putting together any electronic parts, you might find it useful to understand one or two underlying principles first.

Even if you have never worked with electronics before, you may already know some basics without realizing it. You'll know from magnets, for examples that like poles repel and opposite poles attract, and the same principle is applied to the way electronic circuits work.

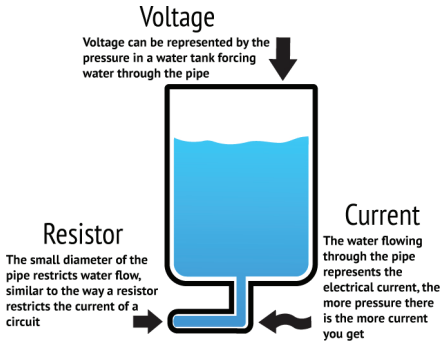
WHAT IS AN ELECTRIC CIRCUIT?

All electronic devices basically work in the same way an electronic circuit is created, which causes a current to flow through various components. To create a circuit, you need a power source (such as battery), a conductor (the wire that carries the electricity from one place to another) and a load (the device the electricity is powering, such as light bulb). The circuit may also include a control device, a simple switch. When the switch is closed, it will connect the circuit to let electricity flow, when it is opened, it will break the circuit to stop the flow of electricity.



MEASURING AND CALCULATING

In any electrical circuit, there are three important factors in play – current, voltage and resistance – and each one has direct relationship with the other two. As we said above, current is the flow of electricity, it is measured in amps (A), voltage measured

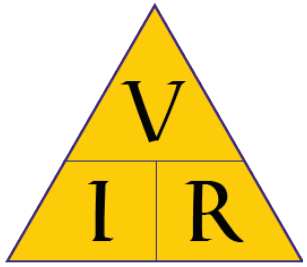


in volts (V) is the pressure of the electrical flow. Resistance (R) is measured in ohms or by symbol Ω, is the measure of how difficult or easy it is for electricity to flow through the circuit. The relationship these three share is illustrated in the diagram.

The key thing to understand is that voltage, current and resistance are always directly proportional to each other, and this is essentially what ohm's Law is. Ohm's Law states that current is equal to voltage over resistance and is often represented by the formula I = V/R, where 'I' is the current in amps, 'V' is the voltage in volts, and 'R' is the resistance in ohms.

The same formula can also be written as V = I x R or R =V/I, meaning that it is always possible to calculate one of the values as long as you know the other two. Likewise, if you make a change to any of the three values in a circuit, you can easily calculate the effect it will have on the others. Have a look at the triangle diagram on the left. This will help you remember how Ohm's Law works.

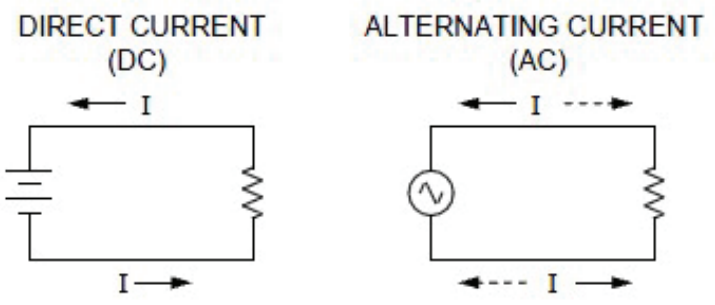
Don't worry – we won't be testing you on this later! But, should you



wish to continue working with electronics beyond our Playground Starter kit, you will need to refer back to Ohm's Law to help you built complex circuits.

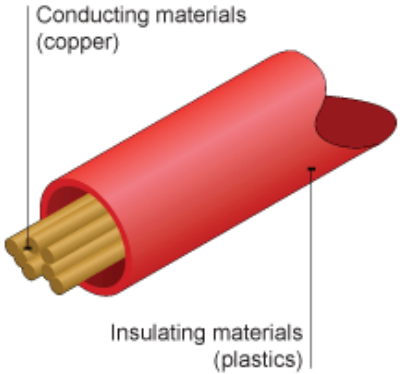
AC VERSUS DC

There are two types of electric current – direct current (DC), which is the type usually supplied by a battery and alternating current (AC), which is the type supplied via a mains socket. The main difference is that with DC, the current flows in one direction only, while with AC, the current continually switches between flowing in one direction and the other. AC can be very dangerous indeed. So we'll be avoiding it altogether in our experiments. All the circuits we will be creating use DC.



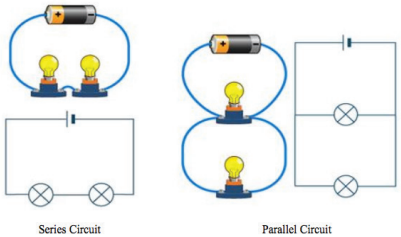
CONDUCTORS AND INSULATORS

Conductors are components that are capable of allowing an electric current to flow freely, while insulators are the opposite – components that block the flow of electricity. Certain types of materials make ideal conductors and insulators. Metals are good conductors. Copper is used in most electrical wiring. Plastic, rubber, glass and ceramic, meanwhile, are often used as insulators – mains wiring is always housed in a plastic coating to insulate it.



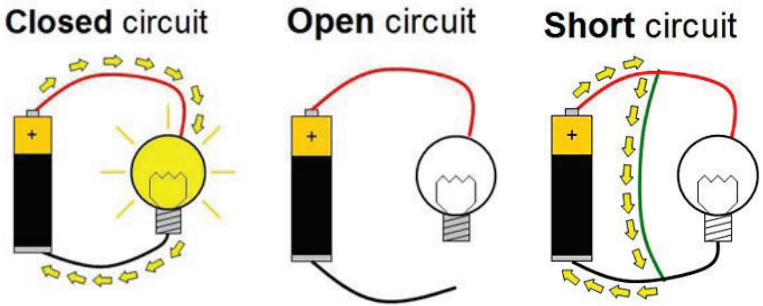
SERIES AND PARALLEL

When your components are connected in a consecutive chain, this is described as 'in series'. In series, the current can only flow in one straight path. When components are connected parallel to each other, this is called 'in parallel' and the current will flow in several paths.



OPEN AND SHORT CIRCUITS

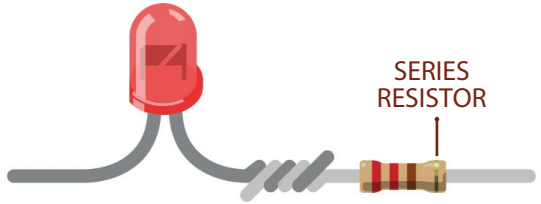
Two of the main common problems with electronics are open and short circuits. Current won't flow at all if the path of the circuit is open at any point, so check your power supply, conductors and components are all connected properly and returned to ground.



A Short circuit, on the other hand, is where the current finds an unintended path through your circuit, potentially bypassing key components, such as resistors or loads. Current will always seek the shortest path through the circuit. But if there's little or nothing to resist current flow, this will result in a much higher current that intended, which could be dangerous, burning up any components along the way. Many electronic devices have built-in-circuit breakers or fuses that will intentionally create an open circuit and stop all current flows as a safety measure, if it detects a current that's too high.

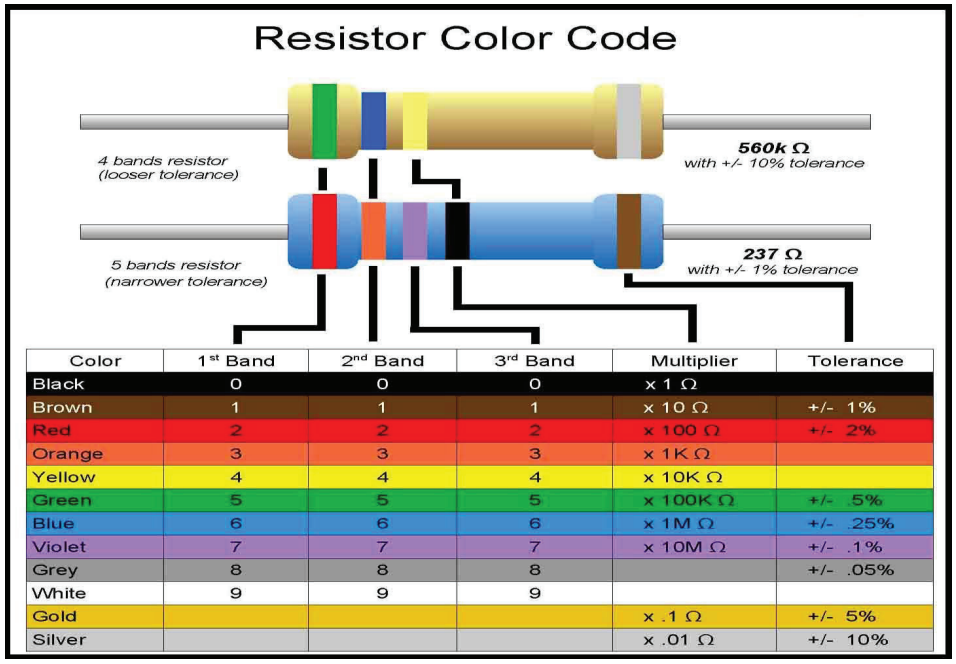
ELECTRICAL RESISTANCE VIA RESISTORS

Resistors are vital components in electronic circuits – they literally resist the flow of current. And why would you want that? Well, basically it's a way of controlling that current. In a simple circuit with 5V power supply and a single LED, for instance, connecting the power directly might pass to much current through the LED and damage it.



Connecting a resistor in series can protect the LED from excessive voltage and current.

EXAMPLE:



Color	1st Band	2nd Band	3rd Band	Multiplier	Tolerance
Black	0	0	0	x 1 Ω	
Brown	1	1	1	x 10 Ω	+/- 1%
Red	2	2	2	x 100 Ω	+/- 2%
Orange	3	3	3	x 1K Ω	
Yellow	4	4	4	x 10K Ω	
Green	5	5	5	x 100K Ω	+/- .5%
Blue	6	6	6	x 1M Ω	+/- .25%
Violet	7	7	7	x 10M Ω	+/- .1%
Grey	8	8	8		+/- .05%
White	9	9	9		
Gold				x .1 Ω	+/- 5%
Silver				x .01 Ω	+/- 10%

You can refer to the Resistor color code table above to identify the correct Resistors for your experiments.

A series resistor must fulfill two tasks: on one hand, it must compensate the voltage difference between the voltage needed to operate the 5 V and the operating voltage of the LED (2.1 V); on the other hand, the amount current the LED can tolerate (0.02 A).

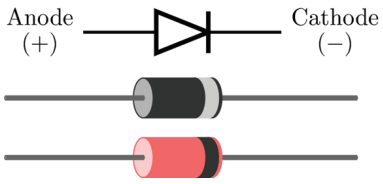
Using Ohm's Law formula to calculate the Resistor value require for the above conditions:

R= U/I
Series resistor (R) = Voltage across resistor / Current
Voltage across resistor (U) = Total Voltage – Operating Voltage of LED
U = 5 v- 2.1 v
U = 2.9 V
R = 2.9V /0.02A = 145 ohms

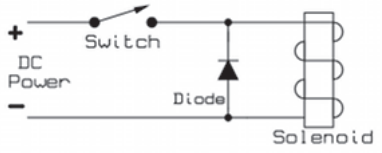
Therefore, under ideal conditions, in this example the resistance should be 145 Ω (ohms). As resistors are sold at fixed values, you can simply use one with a higher value. For LEDs at 5 volts it is common to use a resistor of 220 ohms.

DIODE

The key function of an ideal diode is to control the direction of current-flow. Current passing through a diode can only go in one direction, called the forward direction. Current trying to flow the reverse direction is blocked. They're like the one-way valve of electronics.



If the voltage across a diode is negative, no current can flow, and the ideal diode looks like an open circuit. In such a situation, the diode is said to be off or reverse biased.



As long as the voltage across the diode isn't negative, it'll "turn on" and conduct current. Ideally a diode would act like a short circuit (0V across it) if it was conducting current. When a diode is conducting current it's forward biased (electronics jargon for "on").

LEDS

LED lights up when there is a voltage. However, it is important to connect it properly. Attach the long leg (anode) to the circuit's positive pole and the short leg (cathode) to GND.

LED connected to +5 V and GND lights continuously. You can control the LED by using a digital output pin on the Arduino board which can be programmed to HIGH (+5 V) or LOW (GND). Since the LED may be destroyed at 5V, a series resistor must be integrated in the circuit. At 5V the resistance should be 220 ohms (Ω) (refer to Electrical resistance – Resistor in this chapter)

WIRE COLOR

In the circuits throughout our experiments, we will be sticking to a specific color coding for the wires we use. In each experiment, we will be using red wires for power and black wires for ground, as these are generally considered the standard colors associated with these concepts.

For other types of connections, we will mainly use yellow, blue and white wires in our experiments, for no other reason other than for clarity and continuity. In truth, the colors you use yourself would not matter in the slightest, though using similar color for similar types of components – all orange wires for resistors, for example – can help you recognize and learn about how the various types of connection work.

USING A MULTIMETER

This is a single device that lets you measure voltage, current and resistance. Multimeters have a wide variety of uses – you can test whether a battery is dead or not, for example, and you can measure the current around your circuits.

Multimeter also provides a quick and easy way to work out the value of a resistor. The color coded bands on the resistors are supposed to let you know their value, but they are hard to see and you as a beginner will never remember one from the other.

“The important thing is not to stop questioning. Curiosity has its own reason for existing.”

Albert Einstein

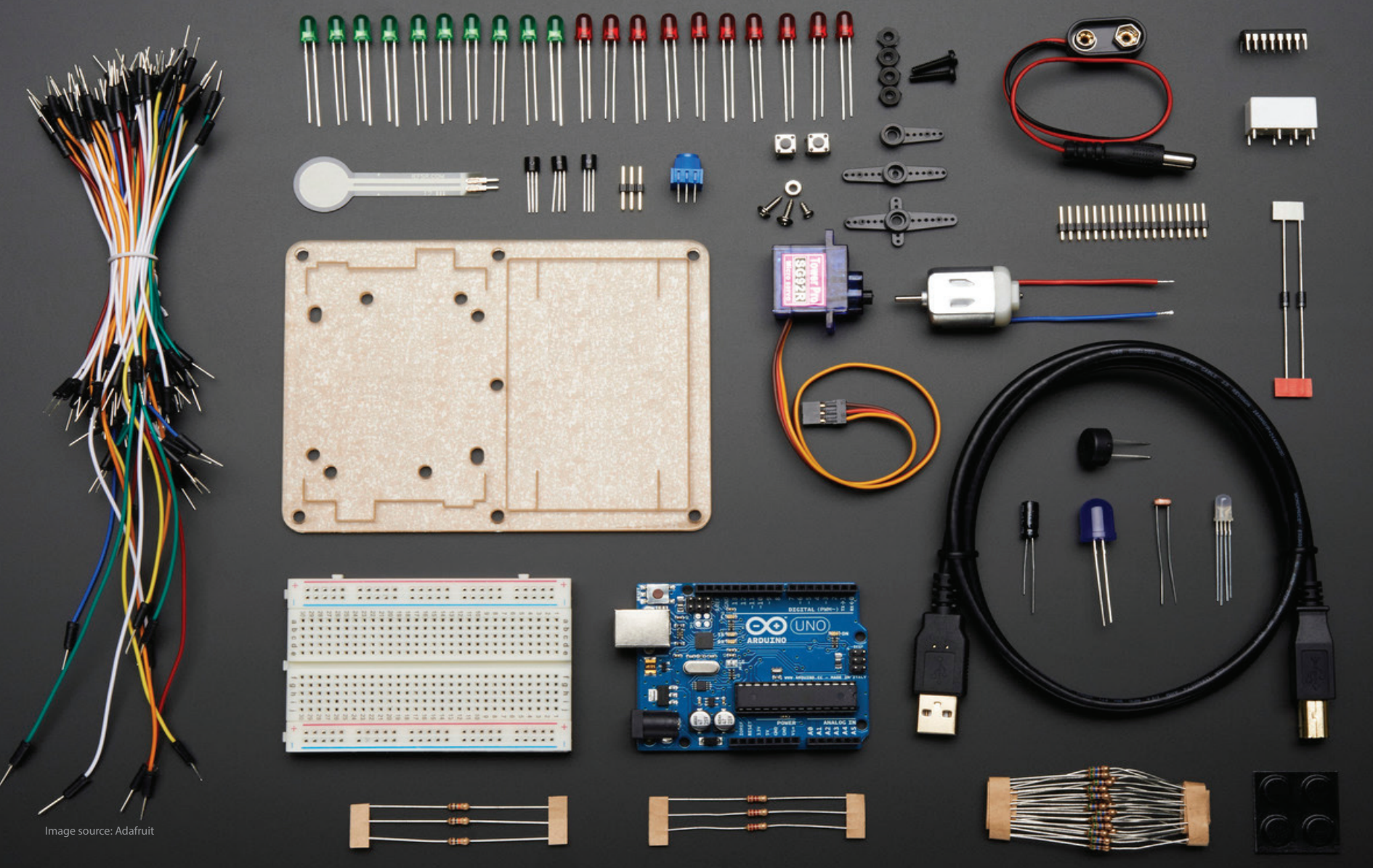
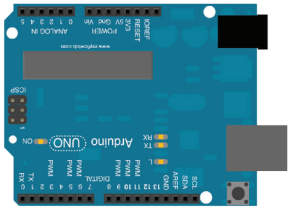


Image source: Adafruit

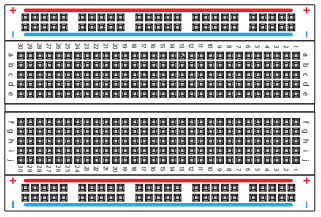
CHAPTER 3

Exploration kit
Base Unit Components

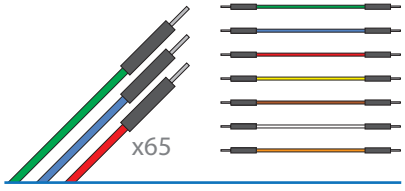
Arduino Uno Board
x1



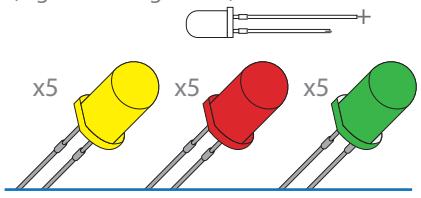
Breadboard
x1



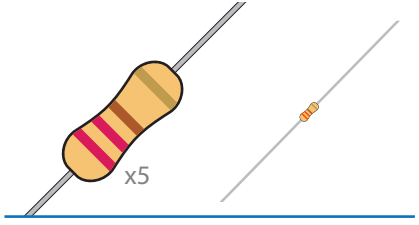
Jumper Wire
Various Colors



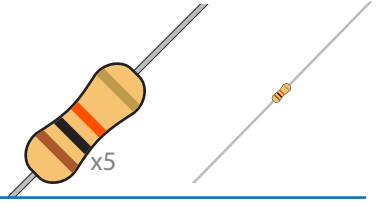
LED (5mm)
(Light Emitting Diode)



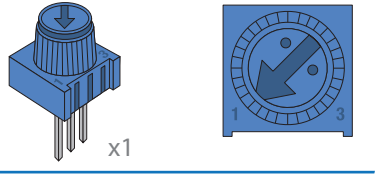
220Ω Resistor



10KΩ Resistor



Potentiometer



RGB LED (5mm)

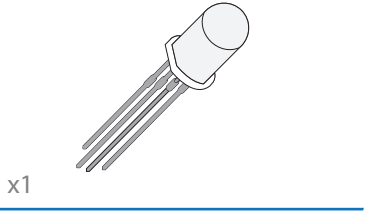
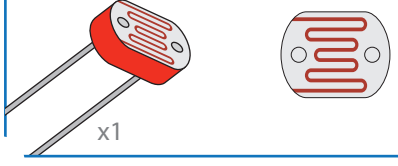
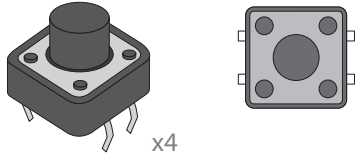


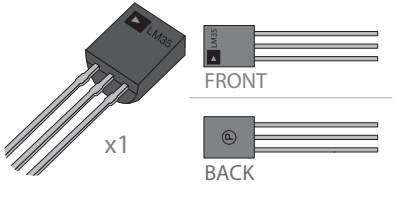
Photo Resistor



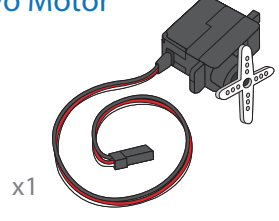
Push Button



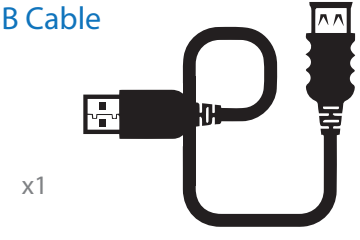
Temp. Sensor
(LM 35)



Servo Motor
SG90



USB Cable



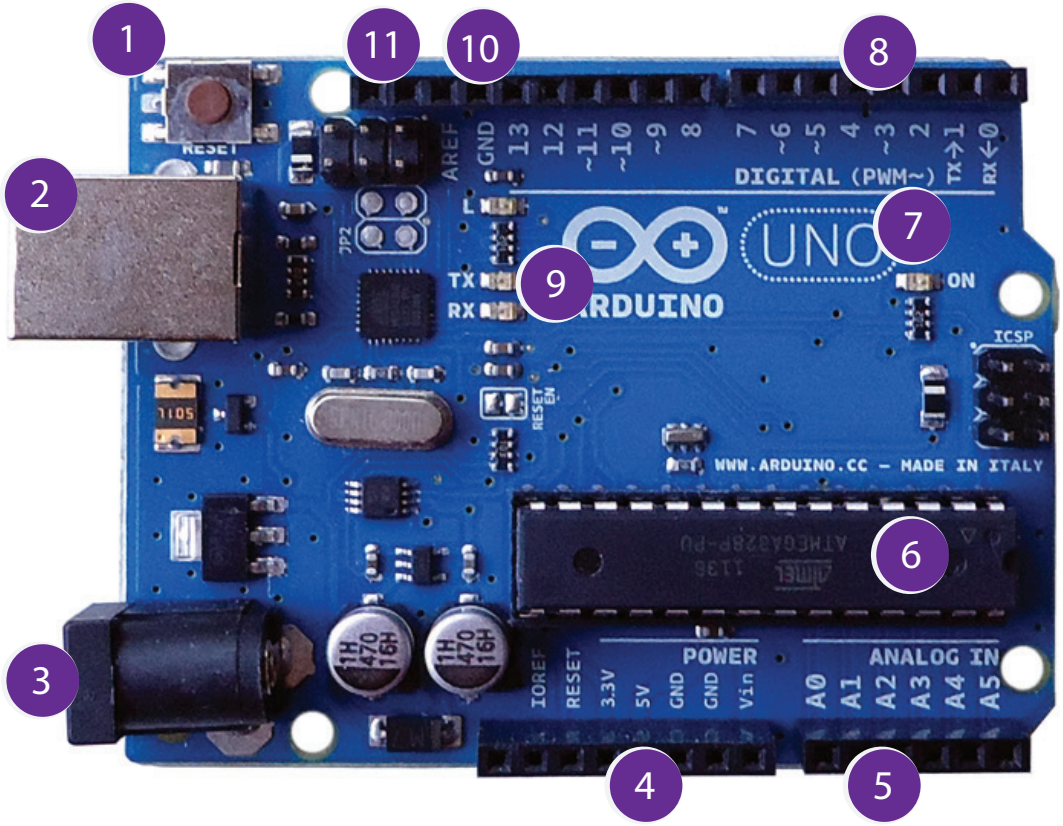
Proximity Sensor

CHAPTER 4

Introducing Arduino

Once the board is loaded with FlowLogic 6 companion firmware, it will configured with following input/output control:

- Four(4) Digital Input
  - Pin 2 ,3 ,4, 5
- Five(5) Digital Output
  - Pin 6 ,7 ,8 ,9
- Four(4) Analog Input
  - Pin A0, A1, A2, A3
- Two(2)PWM Output
  - Pin 6 , 9 and 10
- Servo motor output
  - Pin 12 and 13
- DHT11 Sensor
  - Pin 5
- Tone
  - Pin 9
- Shift Register
  - Pin 8,9,10,
- 16x2 LCD panel
  - Pin SCL,SDA 0r A4,A5
- Robotic Projects
  - Pin 8,9,10,11,12,13



Arduino UNO R3 Board

Quick Start Guide

1. Reset Button
   This lets you reboot the Arduino and restarting the loaded firmware without having to disconnect the board from a power source.

2. USB port
   Enable Connection between Arduino board and the PC that runs FlowLogic 5 application software.USB also supply power (5V and GND) to the Arduino board.

3. Power socket
   An external DC power supply, such as a battery pack or an adaptor, can be attached to power the board when it is not powered by a PC.

4. Power pins
   These are used to provide either 5V or 3.3V of power to the other components in your circuits. There is also a GND pin here to ground your circuit.

5. Analog input pins
   These sic pins can read a constantly varying signal from an analog input (such as a temperature sensor) and convert it into a digital value.

6. Microcontroller
   The microcontroller is the Arduino Uno's main integrated circuit (IC); the brains of the board. The Arduino Uno uses an Amtel ATmega328 chip.

7. Power LED
   This onboard LED illuminates when a power source (including USB) is connected to your Arduino Uno. If it does not light up, it is likely there's a problem.

8. Digital pins
   These 14 pins (numbered 0 to 13) can connect both inputs and outputs. The ones with the title (-) symbol are PWM pins, which can 'simulate 'analog output.

9. TX, RX and L
   TX and RX are LEDs that illuminate to show when the Uno has transmitted (TX) or received (RX) information. L is an onboard LED associated with pin 13.

10. GND pin
    One of several ground pins on the Arduino Uno board. This can be used to ground (in other words, provide a return path for) your circuits.

11. SDL and SDA pin
    Interfacing 16 x 2 LCD Panel via I2C bus

“The true sign of intelligence is not knowledge but imagination.”

Albert Einstein

CHAPTER 5

Introducing FlowLogic 6

Flowlogic 6 is Flowchart based Visual programming software tools that you could download from our website, www.myflowlab.com

Refer to Chapter 7 in this guide to learn how to download it.

You will be using it to learn and develop various application to control devices connected to Arduino board.

Developed for:

For windows 7 and 10

Once you have installed the Flowlogic 6 Application Software, an icon as shown below will appear on your PC desktop.

A. LAUNCHING FLOWLOGIC 6

On Windows PC desktop, launch Flowlogic 6 by clicking on the FlowLogic 5 program icon.

B.THE WELCOME DIALOG

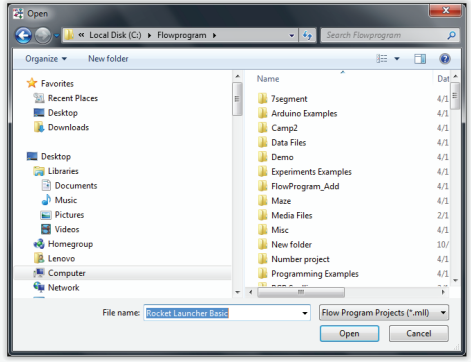
When FlowLogic 6 is first launched, the Welcome dialog box will be displayed. Choose one of the options to get started:

1. Sample FlowProgram
   Select this option to open a dialog window that consists example Flowprogram that you can explore before developing your own.

2. Open FlowProgram
   Select this option to open the file dialog window that displays Flowprogram that you has previous written. Click on the folder and select from the listing to open it. You can also select this option from the File Menu - “Load”.

3. Create New
   Click this button if want to create a new Flowprogram. a workspace will be opened for you and you can start creating.

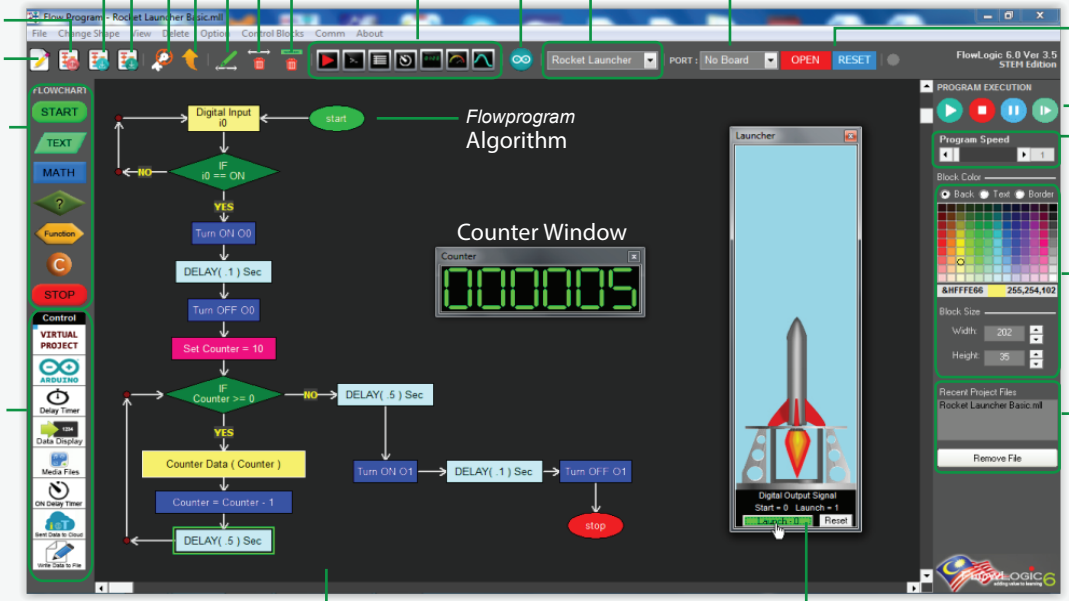
4. Exit
   Clcik this button to exit Flowlogic 6.



Open dialog window

C. FLOWLOGIC 6'S INTEGRATED DEVELOPMENT ENVIRONMENT (IDE) SCREEN

Below are the FlowLogic 6 screen with the Rocket launcher mimic (virtual project) and its control Flowprogram loaded:



D. THE TOOLBAR

On the Toolbar there are various icons and list box for you to select to carry out your required operation as explained below:

1. New
   To empty the workspace for you to construct a new Flowprogram.

2. Open
   To load existing Flowprogram that you have stored in the Storage media.

3. Save
   To store currently loaded Flowprogram into a storage media.

4. Save As
   To store currently loaded Flowprogram with a new file name.

5. Lock / Unlock Workspace
   Allows you to lock and unlock the workspace. To construct a Flowprogram you need to unlock the workspace. There is an indicator on the right bottom corner of the workspace.
   Red – Locked, Green – Unlocked.

Flowprogram Execution Panel


Run the Flowprogram

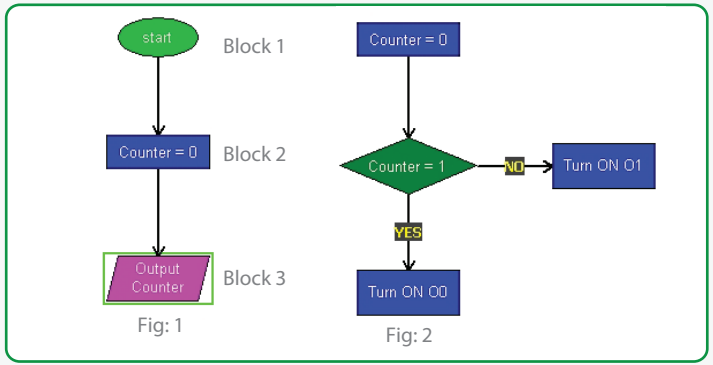

Pause the Flowprogram


Stop the Flowprogram


Continue from Pause

Connect Line

Use the Connect Line tool to connect blocks together and define the flow of the program. All blocks except for a Stop symbol need a line proceeding from them to tell the computer what to run next. Every Decision block needs both a YES and a NO line. To add a line, select the Connect Line tool from the tool bar. Then, click on the block from which you want the line to flow and to the next block that you want to connect. If the block is a decision block, first you need to connect the YES route and then the NO route (Fig: 2)

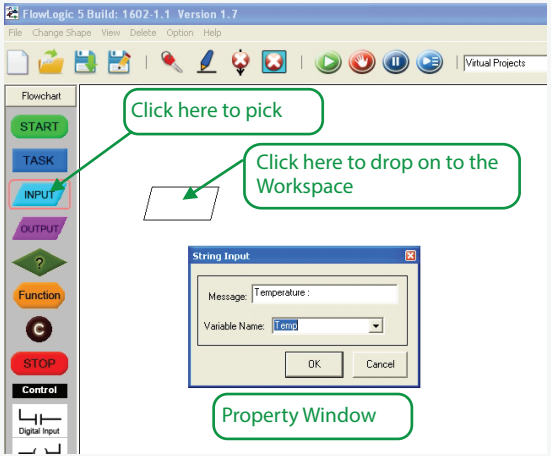


Example: If you want to connect block 2 and block 3 (Fig 1), first click on block 2 first and then block 3 then click the connect line from toolbar or right click on block 3 and select connect line from the popup menu. You can also use the keyboard shortcut Ctrl-C to connect the line.

To Delete line, for example: Line between Block 1 and Block 2, first click block 1 second click block 2 and then click delete line icon from toolbar or use keyboard shortcut Ctr-D.

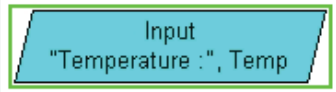
E. HOW TO CONSTRUCT A FLOWPROGRAM

To construct your Flowprogram, pick the command blocks from the command panel and drop onto the workspace. When a command block is placed, a properties window will be displayed. Select or type appropriate value or variables for the your specific requirements.



For example: Flowchart command block – Input. Once you have entered the property contents, click OK on the property window to apply the changes to the command block.

The completed command block will look appear as shown below:



Flowchart Input block

F. CORE COMMAND BLOCKS AND PROPERTIES

FLOWCHART COMMAND BLOCKS

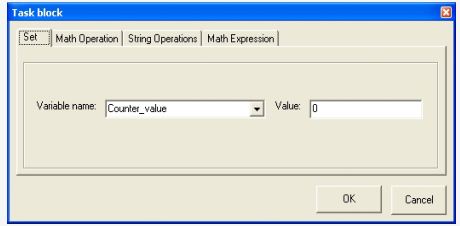
Start / Stop Block

Use the Start block at the beginning of the Flowprogram. Use the Stop block to end a program. Or use the Stop block to end a subroutine.

Task Block

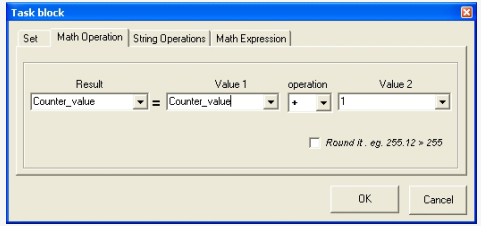
Use Task block when you want your Flowprogram to perform the following requirements:

1. Definition:
Here you could create a variable and assign an initial value to it. The variable must be a string base and the value must be a numeric base.



Example:
Variable name: Counter_value and the Value: 0 (zero)
(Refer to Flowprogram on page 19)

2. Arithmetic Operation:
Here you could create a function for mathematical operation such as Addition(+), Subtraction(-), Multiplication (*) and Division (/).



Example:
Add 1 to the counter_value variable that was defined previously. The function should look like shown:

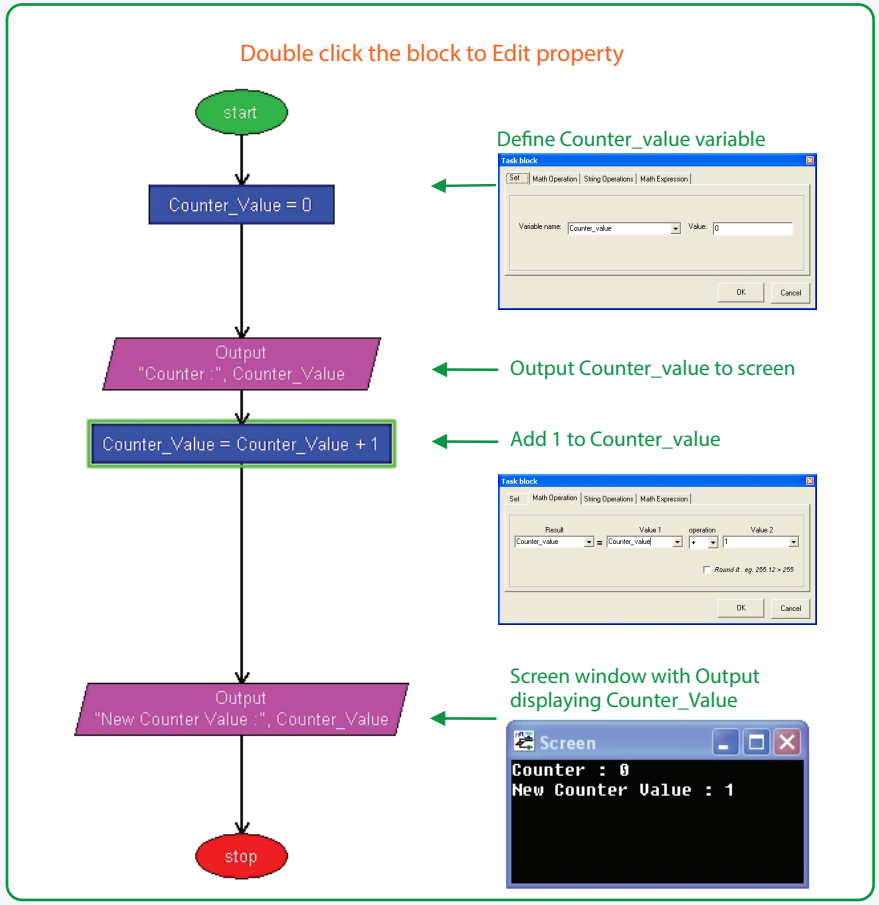
Counter_value = Counter_value + 1

Result = Operand 1 + Operand 2

You could click on the down arrow to display the previously defined variables.

G. EXAMPLE: TASK BLOCK AND OUTPUT BLOCK

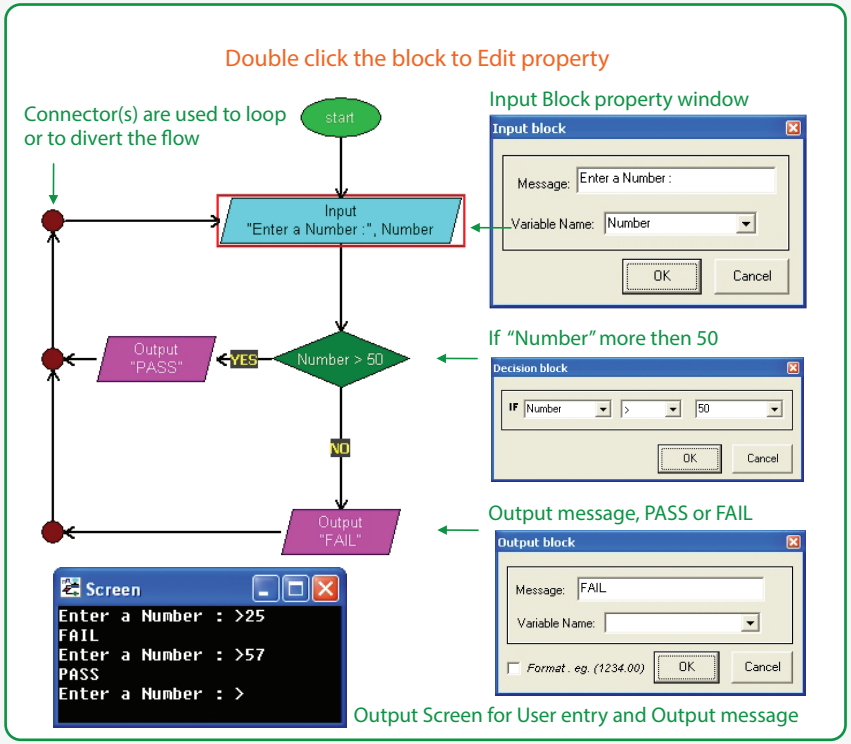
The Flowprogram below illustrate the use of Task block and Output block.



The above flowprogram "Task block example.mll" is available in Example folder.

H. EXAMPLE: DECISION BLOCK AND INPUT BLOCK

The Flowprogram below illustrate the use of Decision block and Output block.



The above flowprogram "Decision block example.mll" is available in Example folder.

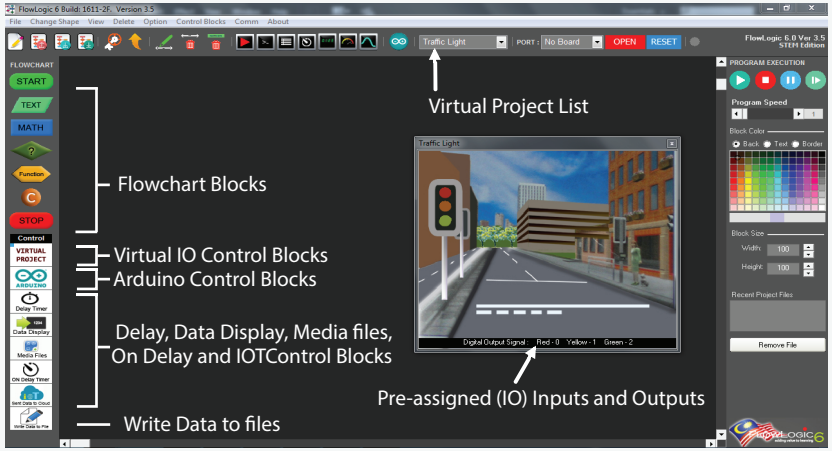
When you run the above flowprogram, the Input block will prompt the screen "Enter a Number :". The value entered will be stored in the variable "Number". The Decision block will then branch the logic flow based on the specified condition.

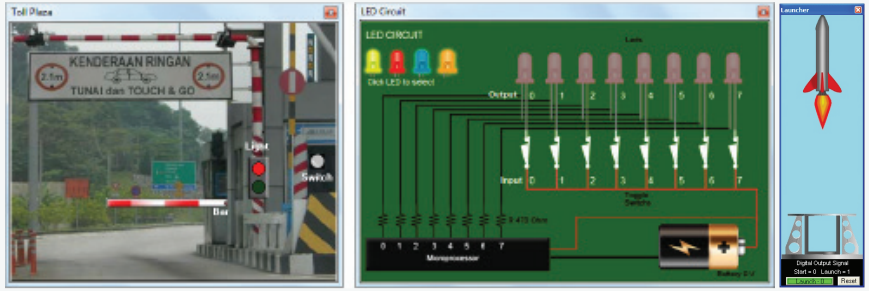
I. VIRTUAL PROJECT

FlowLogic 6 also incorporates numerous pre-built on-screen mimics that we named it Virtual Project, that allows progressive and introductory to programming concepts in manageable steps. You can build programs to animate and manipulate these virtual projects by building Logic Sequences using various command elements.

All the Input/Output control and animations elements have been pre-assigned. You have to refer to this asignments when creating flowprogram to execute this projects.

To access these virtual projects, refer to the drop down list on the toolbar and click the list to display it on the workspace. To create flowprogram to control and animate Virtual project's on screem mimics, you have to use combinations of Flowchart blocks and Virtual Control blocks as shown below.



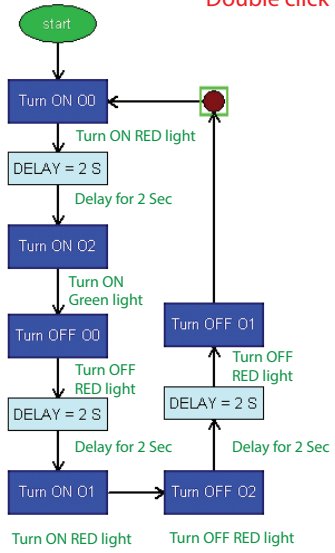


Some of the Virtual Projects: Toll Plaza, LED circuits and Rocket Launcher

FLOWPROGRAM TO CONTROL A SINGLE POLE TRAFFIC LIGHT

Continous loop - Using Digital Output command from Virtual IO control blocks.

Double click the block to Edit property



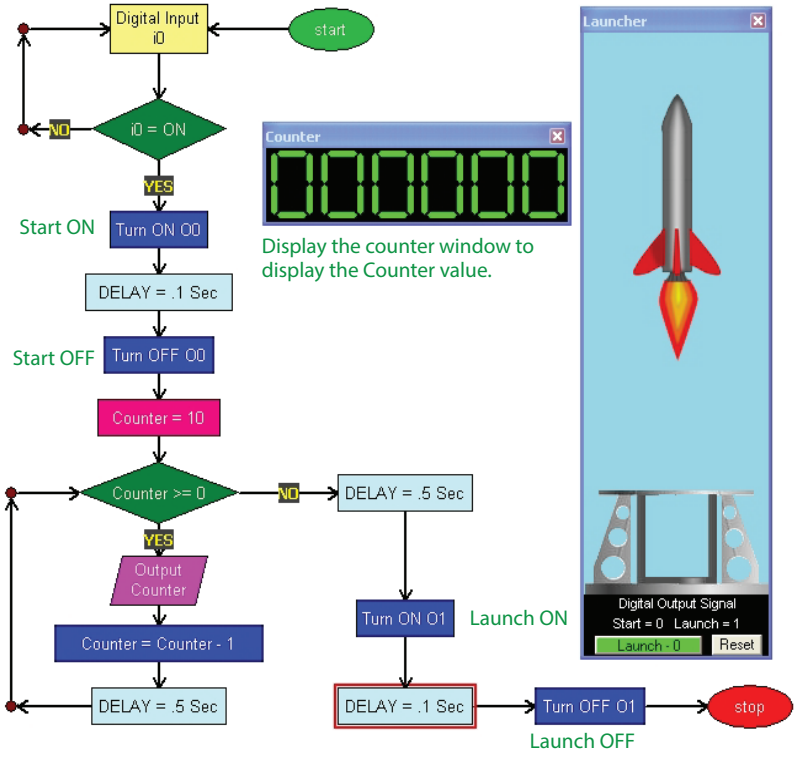
Pre-assigned Outputs

The above flowprogram "Single pole traffic light.mll" is available in Virtual project folder. Display the Trafic mimic from the Virtual project list on workspace and run it.

FLOWPROGRAM TO CONTROL A VIRTUAL ROCKET LAUNCHER WITH COUNT DOWN MODULE

Single loop - using Digital Input and Digital Output command from Virtual control blocks

Double click the block to Edit property

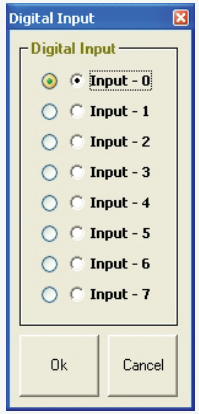


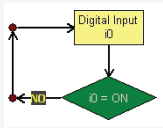
The above flowprogram "Rocket Launcher.mll" is available in Virtual project folder. Display the Rocket Launcher mimic from the Virtual project list on workspace and run it.

J. VIRTUAL CONTROL BLOCKS

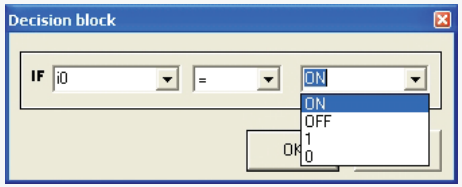
1. Digital Input Block Property

Select the Digital Input option as assigned and click OK.





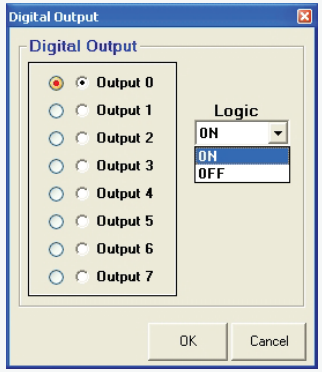
An digital input block need to be scaned until the condition is ON or OFF.

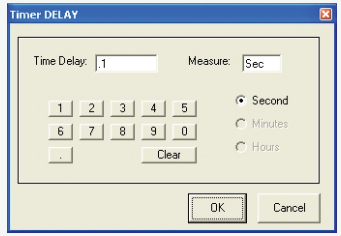


Use the Decision block to define the condition. Click the down arrow to select from the list.

2. Digital Output Block Properties

Select the Digital Output option as assigned, define the logic and click OK.





Timer delay command is used to pause between execution of blocks.

K. ON DELAY-TIMER

Unlike DELAY Timer, ON DELAY Timer command block will not pause the flowprogram execution.

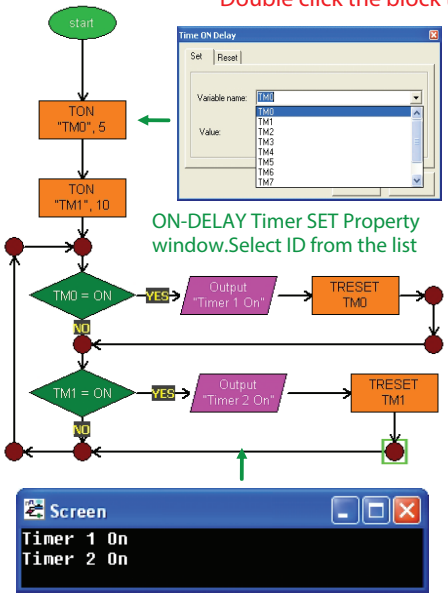
The ON DELAY Timer operates in the background. You need to SET the timer pre-defined ID (TMO,TMO...) with value for it to operate. When the set Timer ID reaches the set value, it will change the status of that ID to ON. You could then use the status of the ID to carry out follow up action. The RESET command will change the status to OFF else the Status will remain ON.

There are 10 pre-defined ON DELAY Timer ID, TM0 to TM9. Select them from the pull down list to SET and RESET the timer ID.

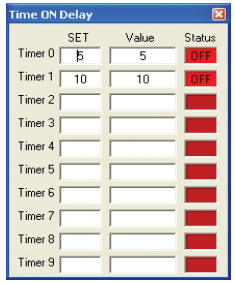
FLOWPROGRAM TO ILLUSTRATE ON DELAY TIMER

Continous loop - Using Digital Output command from Virtual control blocks.

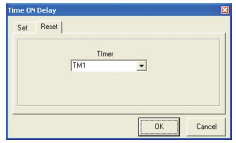
Double click the block to Edit property



ON-DELAY Timer SET Property window.Select ID from the list



ON-DELAY Timer monitor window. Access via toolbar

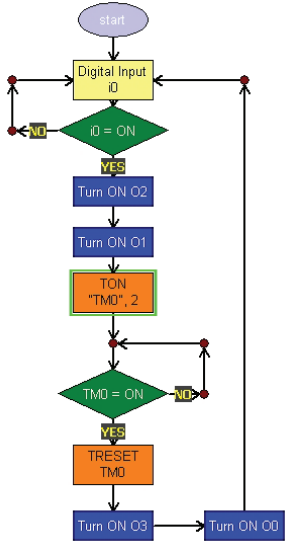


ON-DELAY Timer RESET property window.

The above flowprogram "on delay time.mll" is available in Example folder.

FLOWPROGRAM TO CONTROL A VIRTUAL TOLL PLAZA

User Input - Using Digital Input, Digital Output and ON DELAY Timer command from Virtual control blocks.



When the flowprogram running - click ON / OFF the switch to control the bar.

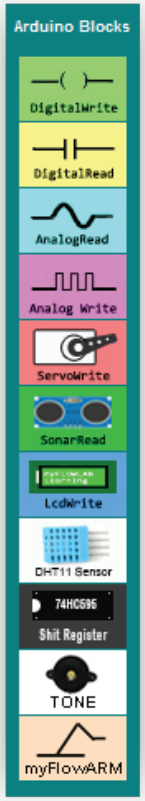
The above flowprogram "Toll_Plaza.mll" is available in Virtual Project folder.

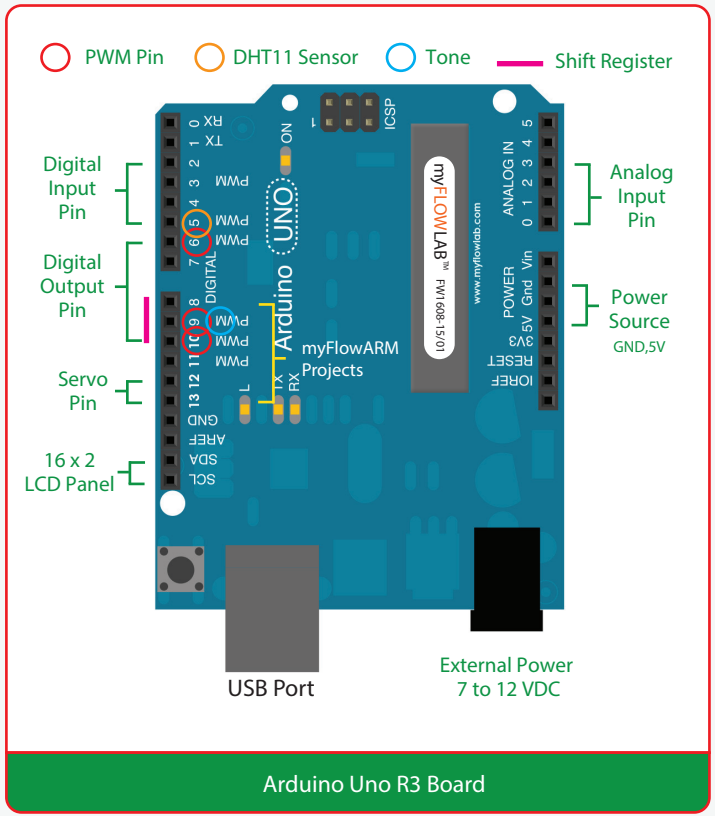
L. ARDUINO CONTROL BLOCKS

Arduino Control Blocks is used when you want to control and monitor devices connected to the Arduino board that comes with myFlowLab Exploration kit.

These command blocks will access your PC USB port to communicate with the Arduino board. So it is necessary that you have connected the Arduino board to your PC USB port when running flowprogram that uses the Arduino Control blocks.

The following blocks are available for you to use when creating flowprogram to access the Input/ Output Pins on Arduino board as as shown on the board.





Before you could run any flowprogram that uses Arduino Control Blocks, you must first open the communication port. When you launch Flowlogic 6, it will automatically detects if any board connected. If there is, a communication port name will be dispalyed as example shown below.

To open the port, click the OPEN button. To close click the CLOSE button. Click RESET button to Reset the communication port.

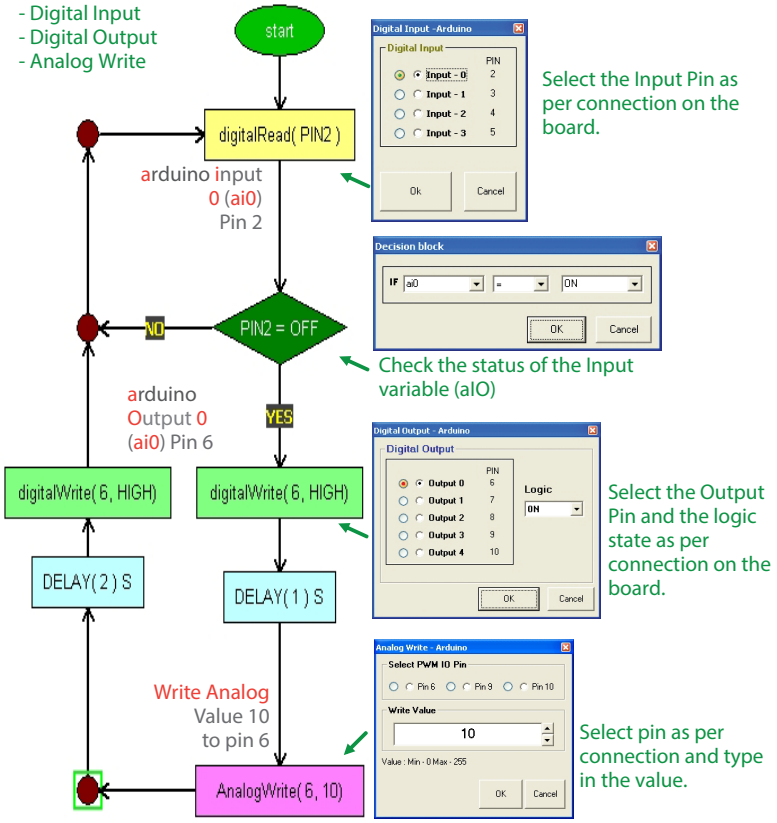
M. ARDUINO CONTROL BLOCKS PROPERTY

REFER TO CHAPTER 10 "EXPERIMENTING" FOR ARDUINO BOARD CONNECTIONS

EXAMPLE FLOWPROGRAM TO ILLUSTRATE THE USE OF THE ARDUINO COMMAND BLOCKS

Command Blocks
- Digital Input
- Digital Output
- Analog Write

Double click the block to Edit property

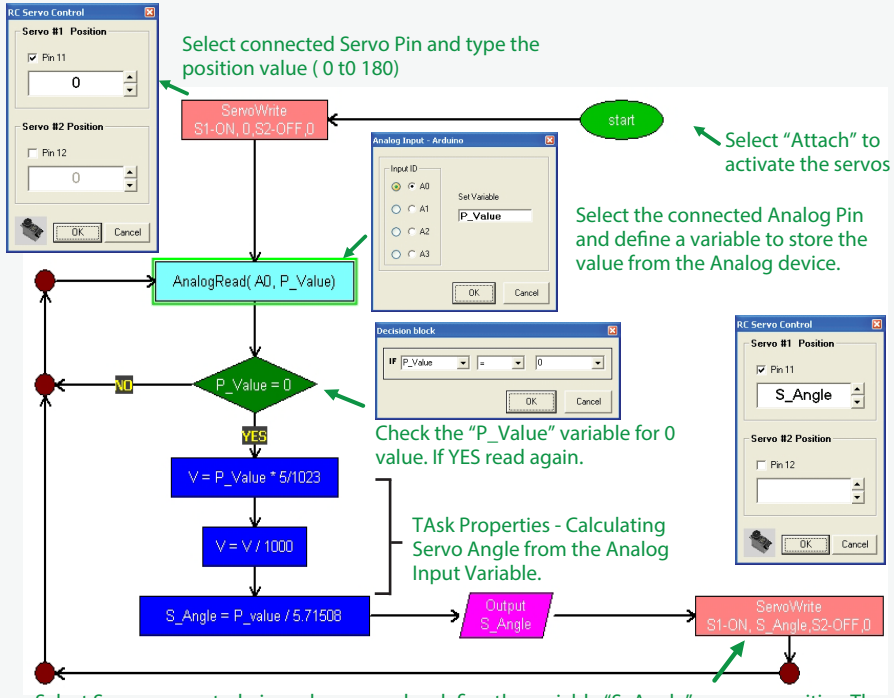


"Arduino Property 1.mll" is available in Example folder.

EXAMPLE FLOWPROGRAM TO ILLUSTRATE THE USE OF THE ARDUINO COMMAND BLOCKS

Command Blocks
- Analog Input
- Servo Setup
- Servo control

Double click the block to Edit property



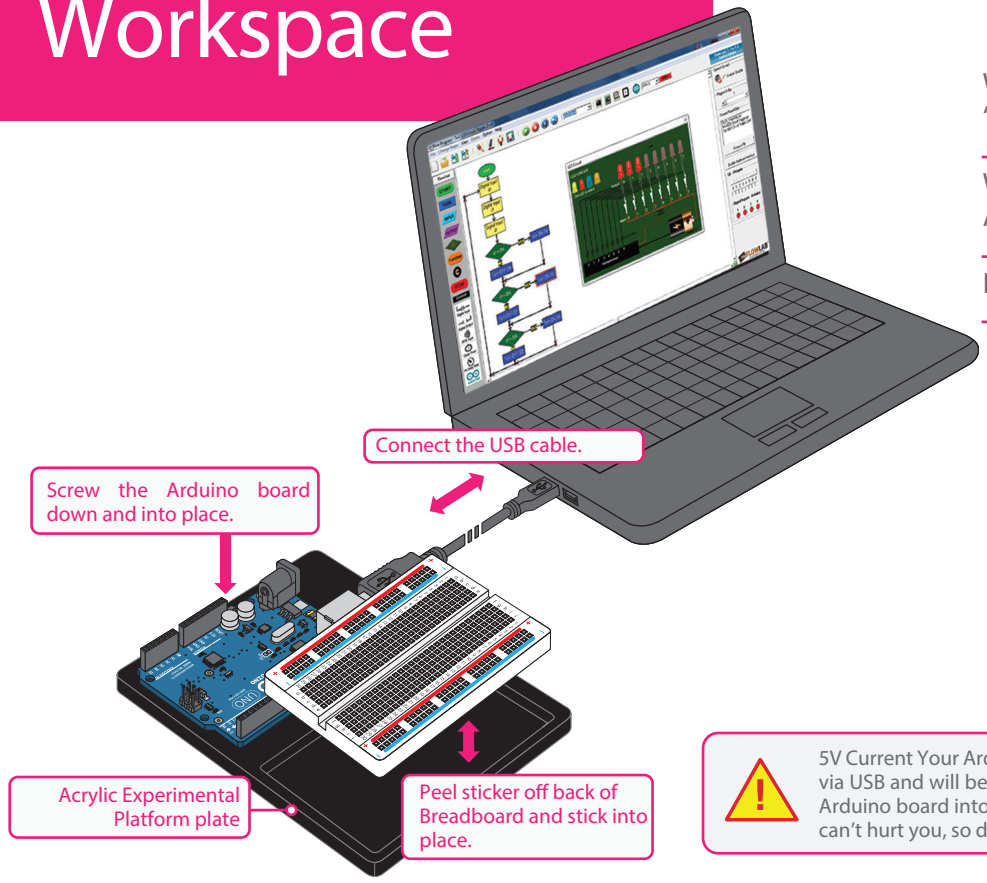
Select Servo connected pin and you can also define the variable "S_Angle" as servo position.The Servo position will defend on the pre-calculated Analog Output value.

"Arduino Property 2.mll" is available in Example folder.

CHAPTER 6

Preparing your Workspace

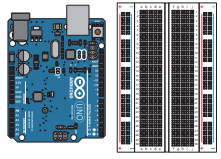
You obviously will require a Table and a PC (Windows 7 or 8 Operating System installed Desktop or Laptop) and myFLOWLAB™ Prototyping board as shown below. You also must ensure you have installed our Flowlogic 6 application software in your PC. The diagram below illustrates how you could setup your workbench for experiments.



When you plug in the Arduino board to your PC, the 'ON' and 'L' led on the Arduino board should light up.

When there is a communication between Flowlogic 5 and Arduino the 'TX' and 'RX' led must light up.

If not, check the Arduino board, USB cable and USB ports.



TIPS

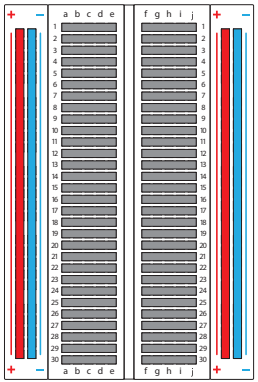
Make sure the text on the Arduino board and Breadboard is facing up so you can read them.

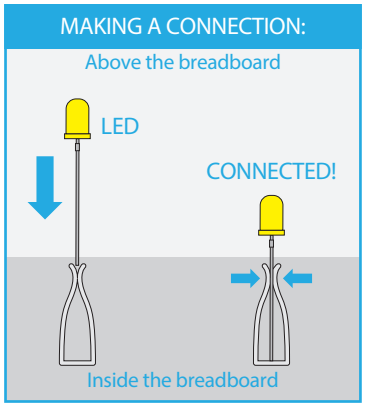
5V Current Your Arduino runs on five (5) volts. This is the power that will be supplied from your computer via USB and will be the driving force behind any components you use in your circuits. By plugging your Arduino board into your computer, you are supplying it with just the right voltage it needs to work. 5V can't hurt you, so don't be afraid to touch anything in your circuit.

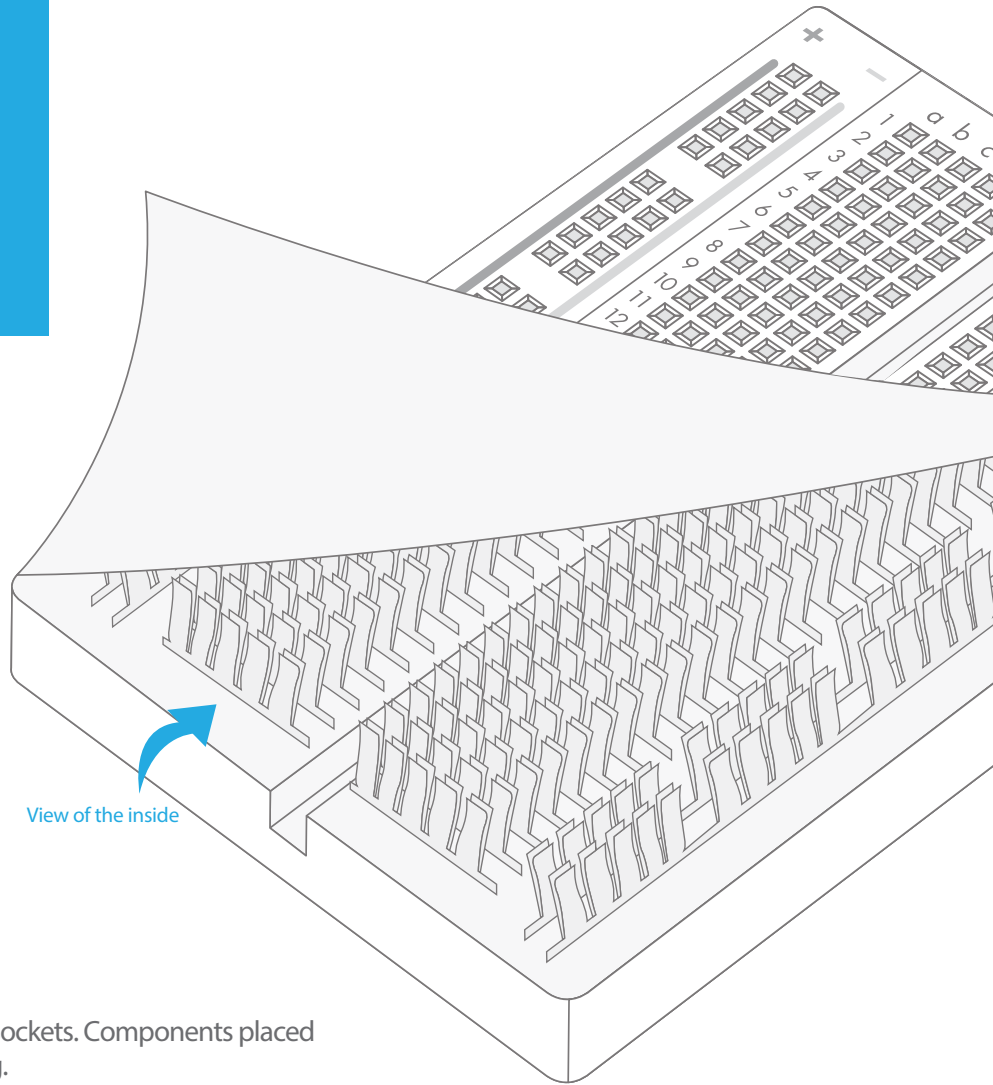
CHAPTER 7

How to use Solder Less Breadboard

HOW'S IT ALL CONNECTED?







+ Power:
Each + sign runs power anywhere in the vertical column.

− Ground:
Each - sign runs to ground anywhere in the vertical column.

Horizontal Rows:
Each of these rows numbered 1-30 are comprised of five horizontal sockets. Components placed in the same row will be connected in a circuit when power is running.

CHAPTER 8

Downloading and Installing Flowlogic 6

Flowlogic 6 with Arduino Control Panel

FlowLogic 6 Ver 3.5 Build 6011

Flowlogic 6 Application Software is available for download via online from our website www.myflowlab.com

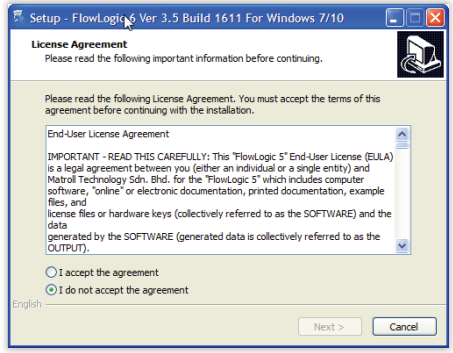
Please ensure your PC fullfills the minimum requirements as listed below:

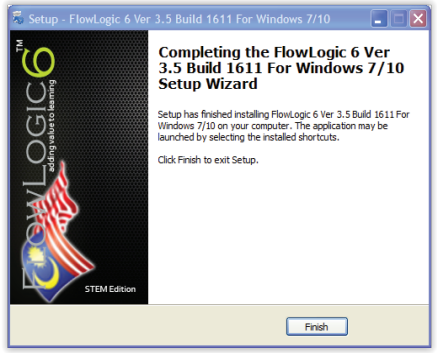
1. CPU: Pentium 750Mhz and above
2. RAM: 1 GB and above
3. Operating System: Window XPX, 7 and 8
4. Harddisk: 1 GB
5. Screen Resolution: 1024 x 768 above
6. USB Port: 1.1 and above

Follow the steps below to download and install it in you PC.

THE STEPS:

1. Goto www.myflowlab.com.
2. Click "Download".
3. Select FlowLogic 6 download button
4. Once downloaded, Click on the downloaded Setup file to install it



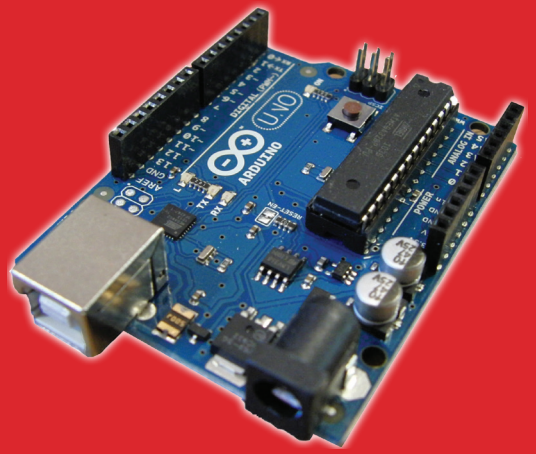


Click the "I accept the agreement" and click Next and follow the instructions complete the installation.

CHAPTER 9

Downloading and Installing Arduino Drivers

for Arduino Board



Arduino Uno board requires a USB driver to interact with. You can download the USB driver from myflowlab.com website and install it or Install it direct from FlowLogic 6 application. Follow the guide below for either options.

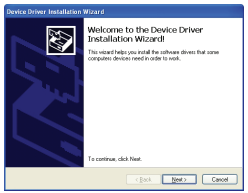
The Driver is provided by Future Technology Devices International Limited “as is” and with disclaimer from any issues that might arrise by using it.

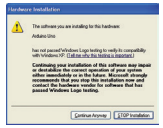
DOWNLOAD OPTION:

1. Go to www.myflowlab.com.
2. Click “Download”.
3. Look for “Arduino USB Driver”.
4. Once downloaded, unzip it into a folder (or default folder)
5. Plug in the Arduino Uno board to PC USB board.
6. Double click either “dpinst-x86.exe” 32 bit or “dpinst-amd64.exe” 64 bit.
   Please check your PC properties to identify which driver is compatible.

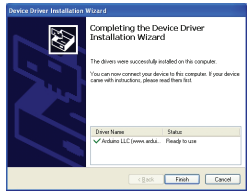
Follow the steps below to complete the installation.

Driver installation Steps





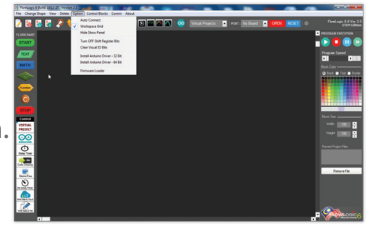
Click Continue Anayway when this window appears



DIRECT OPTION:

1. Launch FlowLogic 6.
   Recommended “Run as Administrator.
2. Select OPTION from menu.
3. Select “32 bit installation or 64 bit installation.

Follow the steps above to complete the installation.



CHAPTER 10

EXPERIMENTING

All of us rush to explore the world of Robotics but the mistake we do is that we tend get into it straight away and fail miserably due to lack of knowledge in many aspects.

To build a Robot requires one to be familiar with Electronic, Sensing and Actuator devices as well as sound knowledge of Microcontrollers and Programming logics.

myFlowab Exploration kit contains all that are required to get you started and the first step guidance to move towards project based learning. From controlling and monitoring a Traffic Light systems to an Autonomous robots and beyond.

In this chapter, you will be exposed to various experiments using beginners level components that are commonly incorporated into Robotics applications and the Programming made easy using Flowlogic 6.

Carry out each experiments in the order it is prepared for you. When you complete each experiment, think out of the box and do it yourself to explore further.

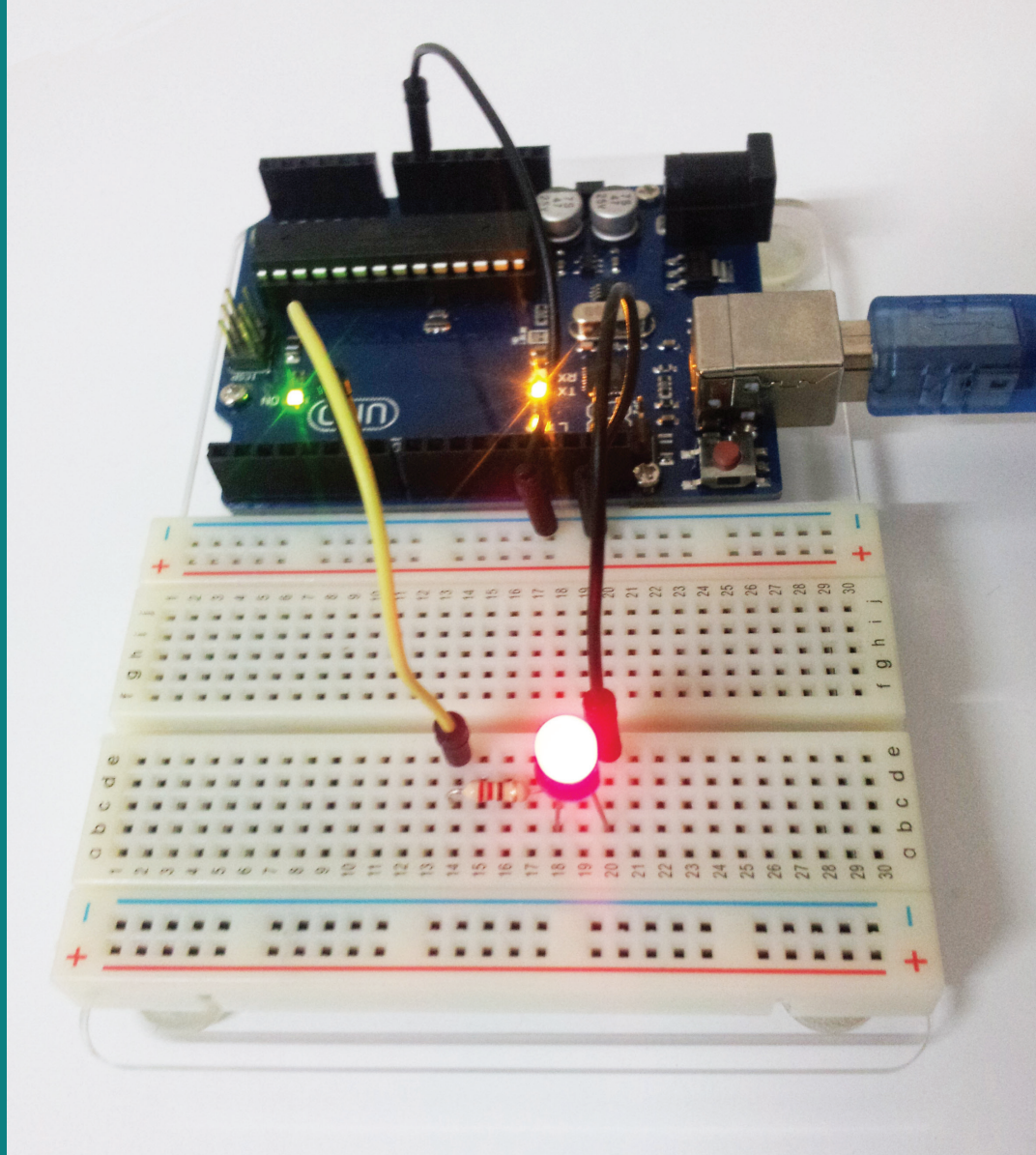
You will be amazed, once you have completed all the experiment in this chapter, apart from acquiring the ICT and Electronic know-how, you will also have enhances your **Computational Thinking**, Problem solving abilities, Critical Thinking, Creativity and Intelligence.

CHAPTER 10A

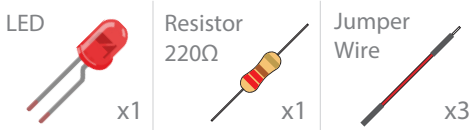
SINGLE LED CIRCUIT

In this experiment, I'm going to show you how to make a simple LED circuit and construct various Flowprograms using Flowlogic 6 to control the LED.

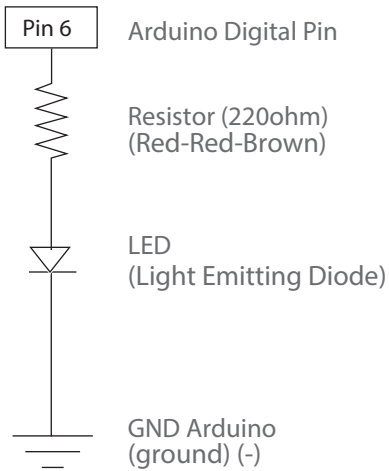
This is a very easy, basic idea, but establishing this important baseline will give you a solid foundation as we work toward more complex experiments.



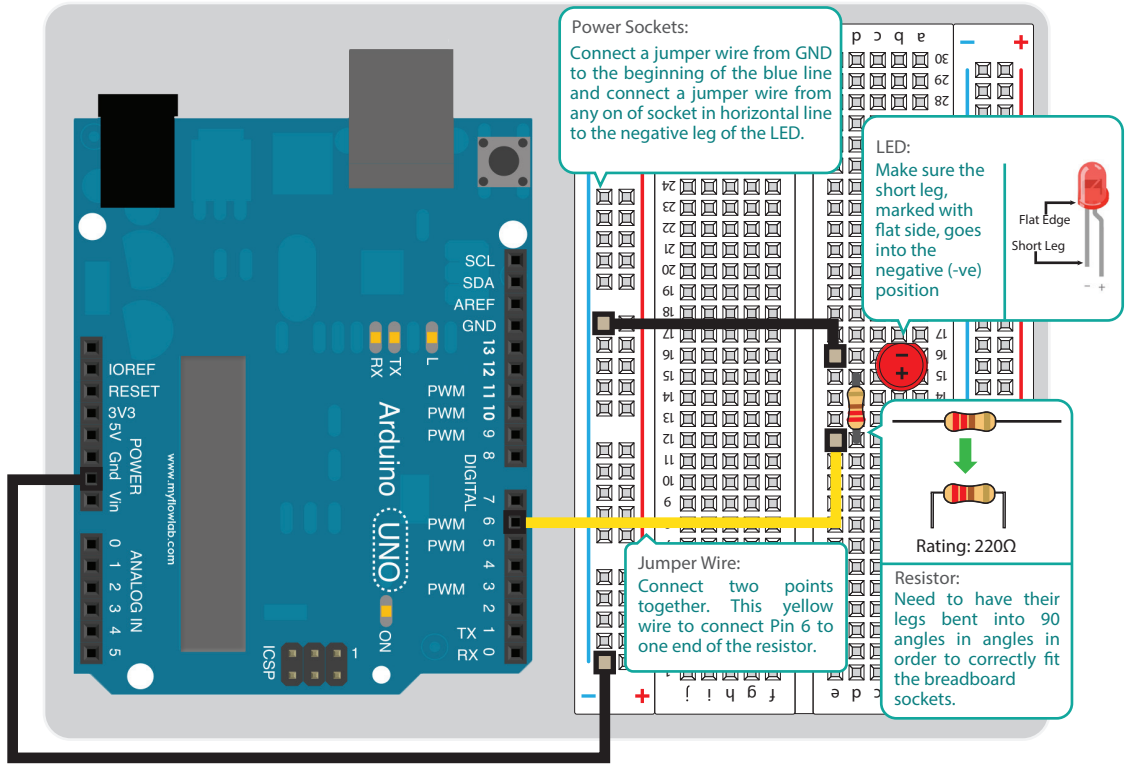
WHAT YOU WILL NEED



SCHEMATIC DIAGRAM



PROTOTYPING CONNECTION DIAGRAM



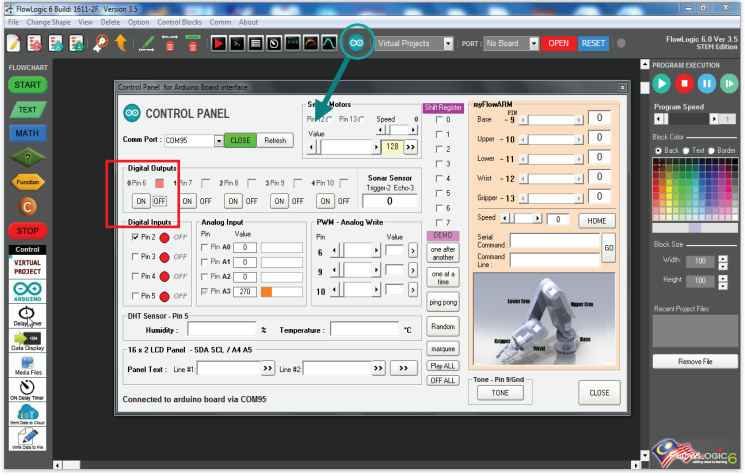
Disconnect Arduino Board USB connection from your PC when carrying out the wires and electronic components connection. Check thoroughly your connections before plugging in the USB cable to your PC.

TESTING YOUR CIRCUIT - SINGLE LED

Launch Flowlogic 6 from your PC and click the Arduino logo as shown below to open the Arduino panel window.

Click the 'Refresh' button to check for connected Communication port. When the Communication port name appears, click OPEN button to open it. The button should turn green if connection is established between your PC and the Arduino board. If it remains Red, check the USB cable connection and the USB port.

Now click on the Digital outputs ON/OFF button 0 to turn ON the LED. If the LED lights up, your connection is correct. If not, please check again your circuit connections.



The Arduino panel is a useful tool you can use to check the entire interface Arduino board provides such as Digital Input/output, Reading and Writing Analog values, manual Servo motor control.

TROUBLESHOOTING
3 THINGS TO TRY TO SOLVE THE PROBLEM

1

LEDs Fail to Light

It is easy to insert the LED backwards. Check the LED leads and ensure they are right way around.

2

Operating out of sequence

Double check that the LED is plugged into pin 6 correctly.

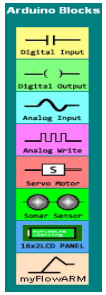
3

Starting Afresh.

Pulling everything out and starting with a fresh slate is often easier than trying to track down the problem.

Please ensure Flowlogic 6 is communicating with the Arduino board. If YES, the TX and RX Led on the Arduino board should blink. If NO, unplug the USB cable from your PC and plug in back.

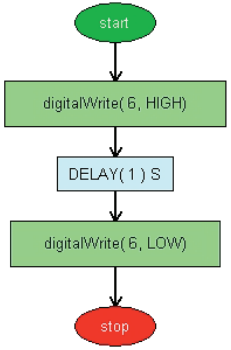
PROGRAMMING EXAMPLES - SINGLE LED



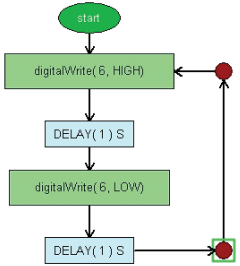
Please ensure you have mastered "Chapter 8: Introducing Flowlogic 6" in this manual before proceeding to this section.

For our Single LED Experiment we will be using the Digital Output and Analog Write command from Arduino Command Blocks. The value range for Command block Analog-Write is 0 to 255 and it is only available for Pin 6 ,9 and 10 on the Arduino board.

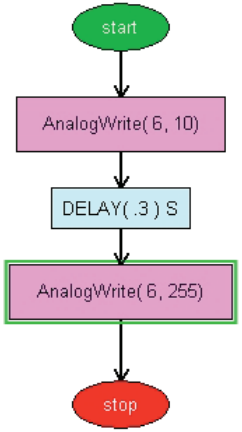
EXAMPLE 1: TURN THE LED ON AND OFF



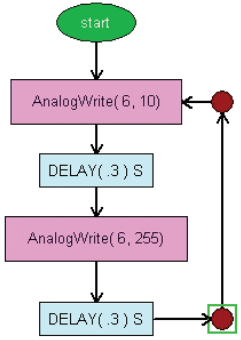
EXAMPLE 2: LED BLINKING



EXAMPLE 3: LED FADE UP AND DOWN

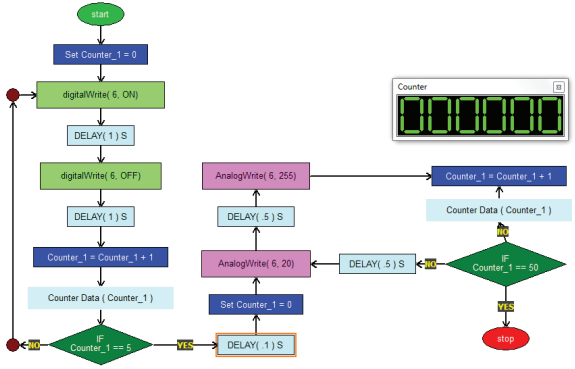


EXAMPLE 4: BLINKING USING FADING EFFECT



EXAMPLE 5: ALARM INDICATOR

Alarm Indicator – In this example the Flowprogram is constructed to blink the LED slower for 5 counts (using Digital Output ON/OFF Technique) and faster for another 50 counts (using Analog Output Fading Technique)

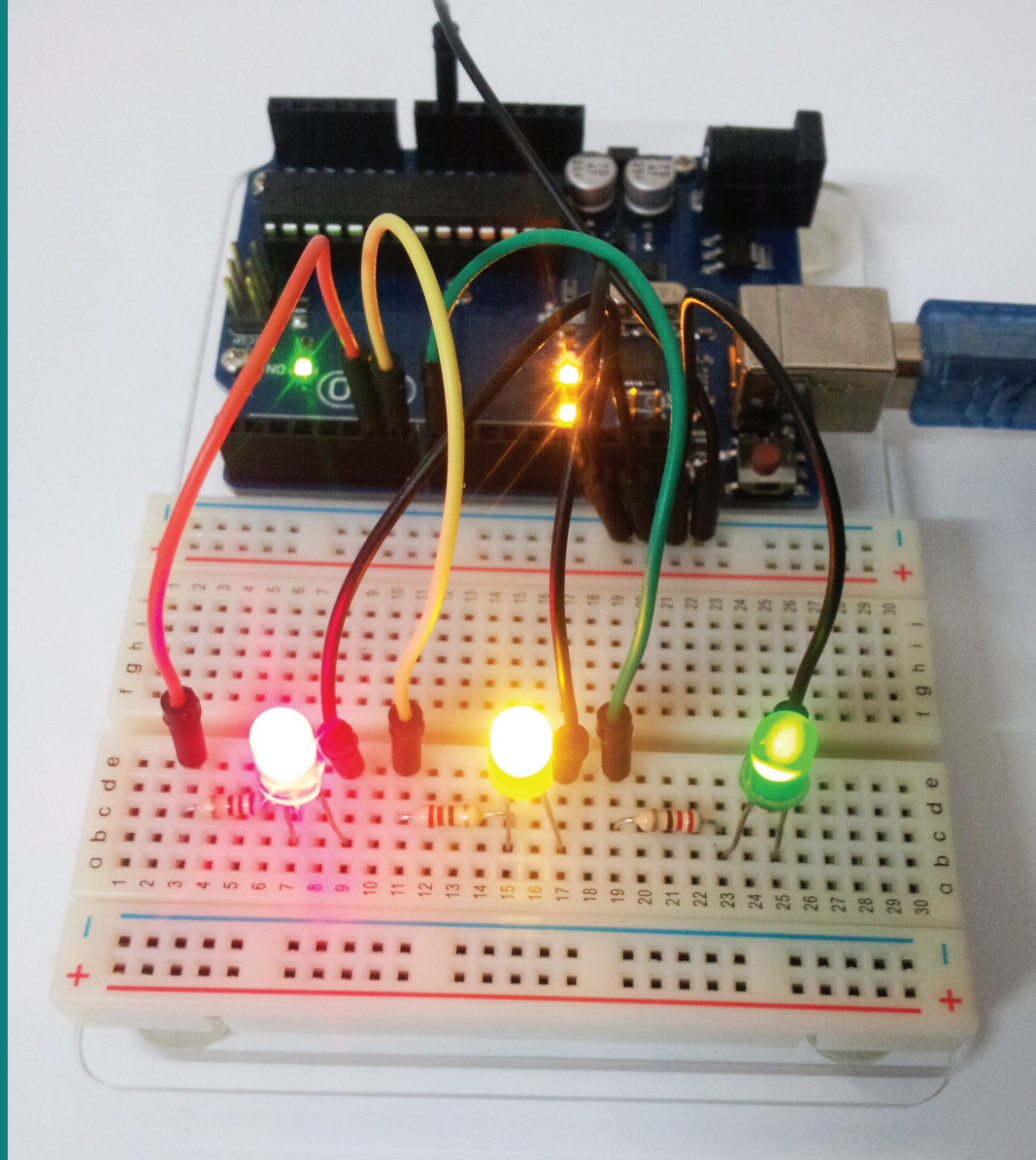


CHAPTER 10B

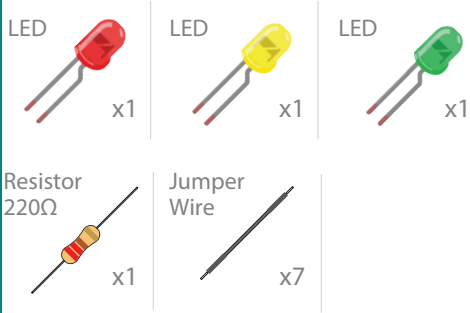
MULTIPLE LED CIRCUIT

Well done !!!, you have completed your first experiment and ready to explore further. You should be familiar with the Arduino board, by the circuit design, protoyping, testing and troubleshooting by now.

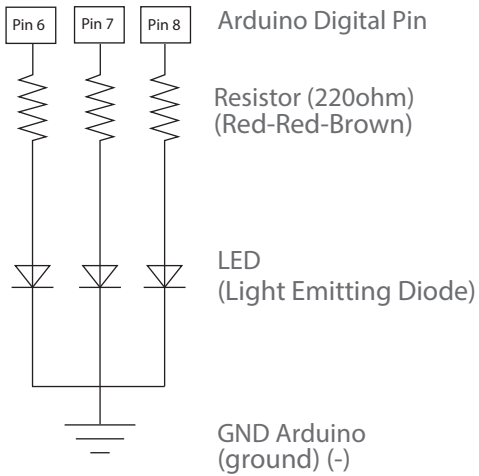
In this experiment, we are going to explore further with LEDs. I'm going to show you how to connect three LEDs (Red, Yellow and Green) and play with it using Flowlogic 6.



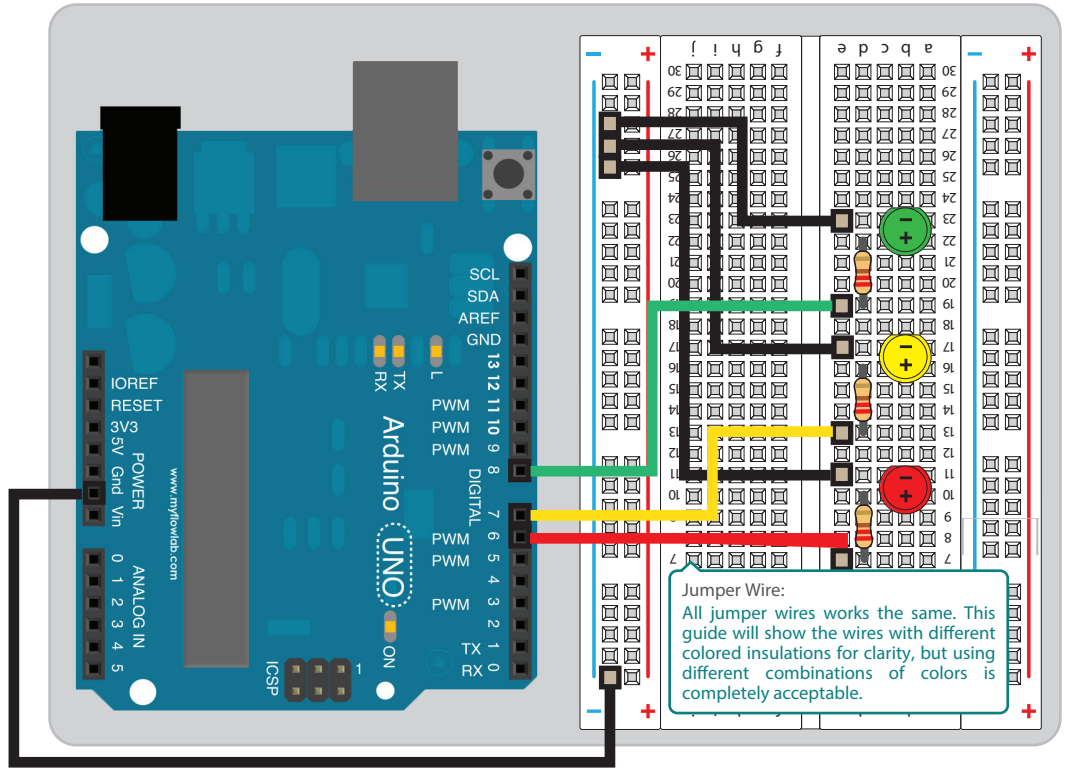
WHAT YOU WILL NEED



SCHEMATIC DIAGRAM



PROTOTYPING CONNECTION DIAGRAM

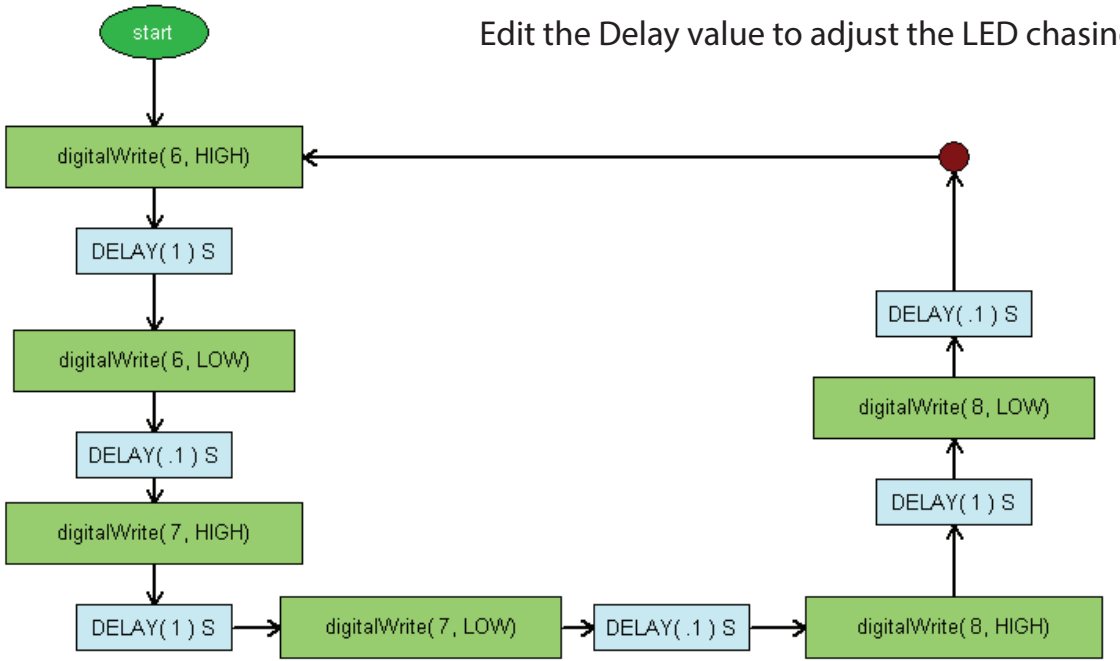


Disconnect Arduino Board USB connection from your PC when carrying out the wires and electronic components connection. Check thoroughly your connections before plugging in the USB cable to your PC.

PROGRAMMING EXAMPLES - MULTIPLE LEDS

Please ensure you have tested your circuit using the Arduino panel tool in Flowlogic 6 and the circuit is working fine before attempting to construct the Flowprogram shown below.

EXAMPLE 1: LED CHASER



Edit the Delay value to adjust the LED chasing speed

This simple Flowprogram contructed to turn on and off the LEDs to simulate a chasing effect.

Please ensure you place a Delay timer block after every Digital output block. You could change the delay property to adjest the chasing effect. It is a close loop sequence, run continously until the stop button pressed.

EXAMPLE 2 : TRAFFIC LIGHT

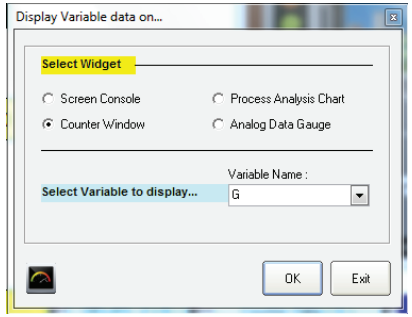
This Flowprogram contructed to control a single pole Traffic light. You can also diplay the Traffic light mimic and counter view panel to output the result on screen. The variable R and G is the variable name , you could use other names if you wish. This variables are used as counter just like in real world application, to delay before turning ON and OFF the light on the traffic light pole. It is a close loop sequence, run continously until the stop button pressed.

Widget's Icon on Toolbar

Data Display block

Traffic Light Virtual Project Mimic

Display the Traffic Light virtual model from the Virtual Project list and the Counter window from the Widget, on the workspace to view the result.



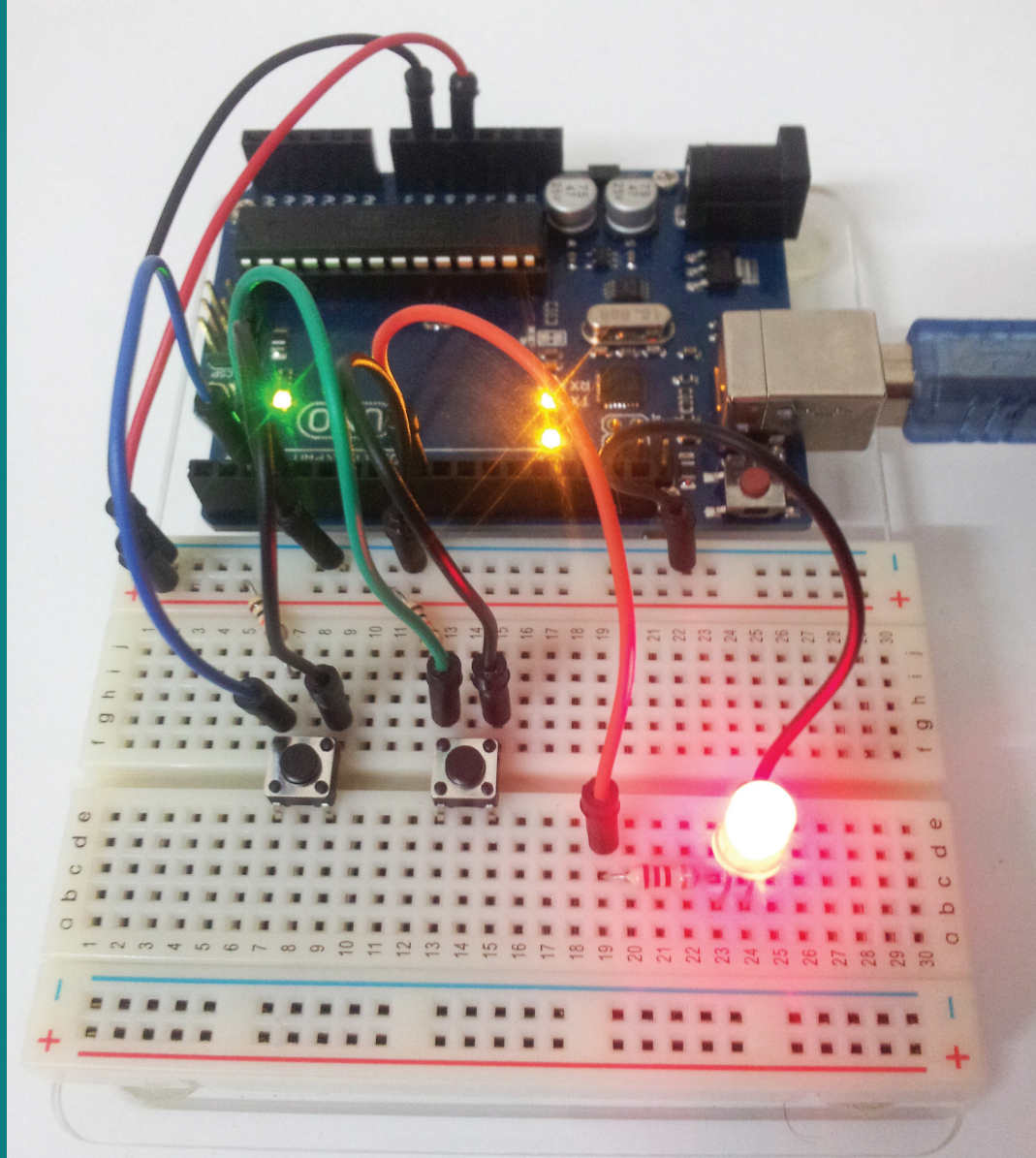
Data Display block property

CHAPTER 10C

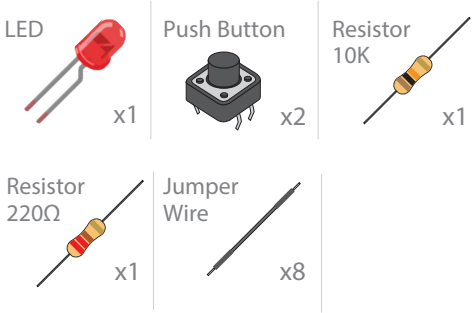
PUSH BUTTON

In this experiment we are going to look at common inputs device, a push button. To build the circuit, you will also use a pull-up resistor, which helps to clean up the voltage and prevents false readings from the button.

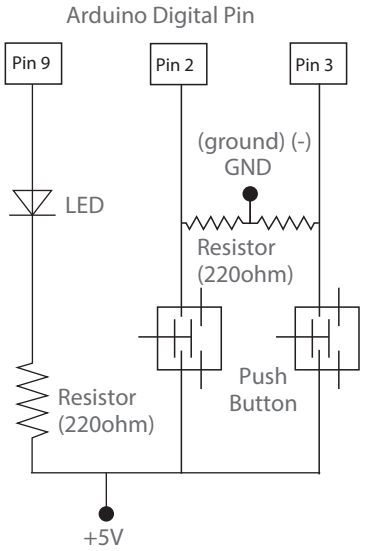
We will also be using a LED in this experiment to turn ON and OFF the LED based on the status of the push button.



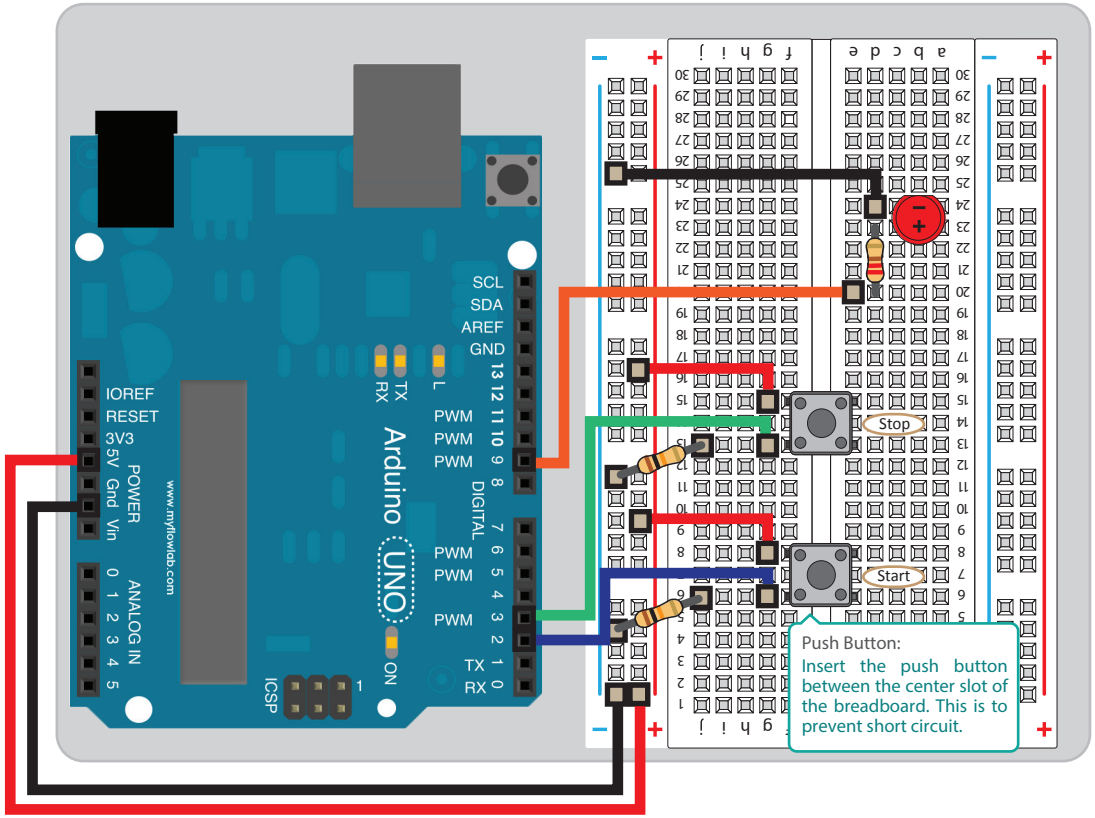
WHAT YOU WILL NEED



SCHEMATIC DIAGRAM



PROTOTYPING CONNECTION DIAGRAM



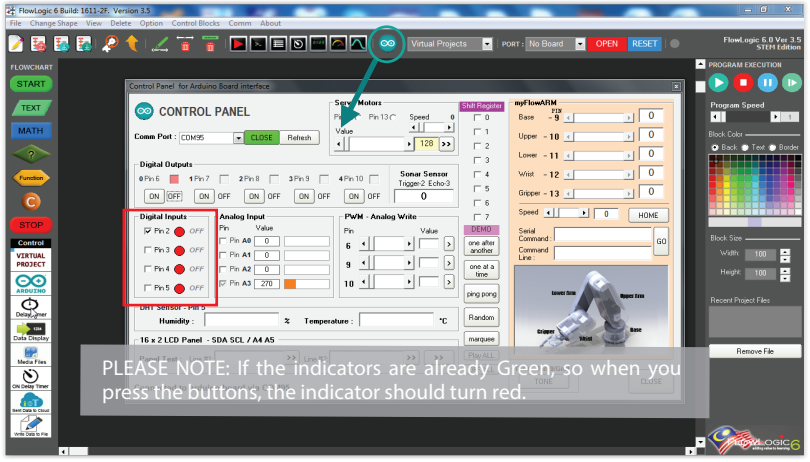
Disconnect Arduino Board USB connection from your PC when carrying out the wires and electronic components connection. Check thoroughly your connections before plugging in the USB cable to your PC.

TESTING YOUR CIRCUIT - PUSH BUTTON

Using the Arduino panel in Flowlogic 6 , test your push button circuit. (Please refer to Test your Circuit of the single LED experiment if you are not very sure).

Since you have connected the push button to input to pin 2 and pin 3, so when you pressed the push buttons, the indictor 0 and 1 on the Digital inputs should turn green (ON-HIGH) or RED (LOW, OFF).

Please ensure the correct Communication port is open when testing your circuit. Click ? button to refresh the communication port.



When the indicator is in Green color, its means the Input signal on that pin is in ON State (HIGH), when the indicator is in Red color, its means the Input signal on that pin is in OFF state (LOW).

You need to take note of the Pin condition and apply it arcordingly during programming.

TROUBLESHOOTING
3 THINGS TO TRY TO SOLVE THE PROBLEM

1

LEDs Fail to Light

The pushbutton is square, and because of this it is easy to put it in the wrong way. Give it a 90 degree twist and see if it starts working.

2

The Arduino panel indicator

Due to the nature of the push button, If it is a Normally closed type, the indicator will be green even you never pressed it.

3

Starting Afresh.

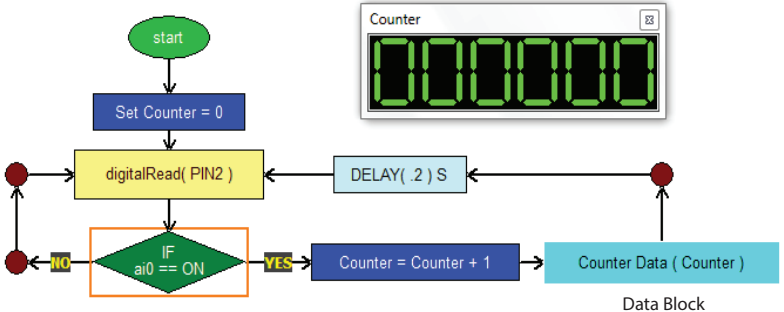
Pulling everything out and starting with a fresh slate is often easier than trying to track down the problem.

Please ensure Flowlogic 6 is communicating with the Arduino board. If YES, the TX and RX Led on the Arduino board should blink. If NO, unplug the USB cable from your PC and plug in back.

PROGRAMMING EXAMPLES - PUSH BUTTON

EXAMPLE 1: PUSH BUTTON COUNTER

This Flowprogram simply increase the Counter value of variable “Counter” by 1 each time the push button connected to Pin 2 (START) is pressed.

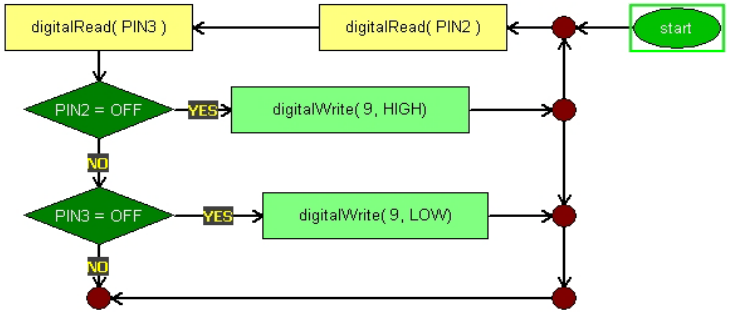


Data Block

To display the value of FU on the Counter window

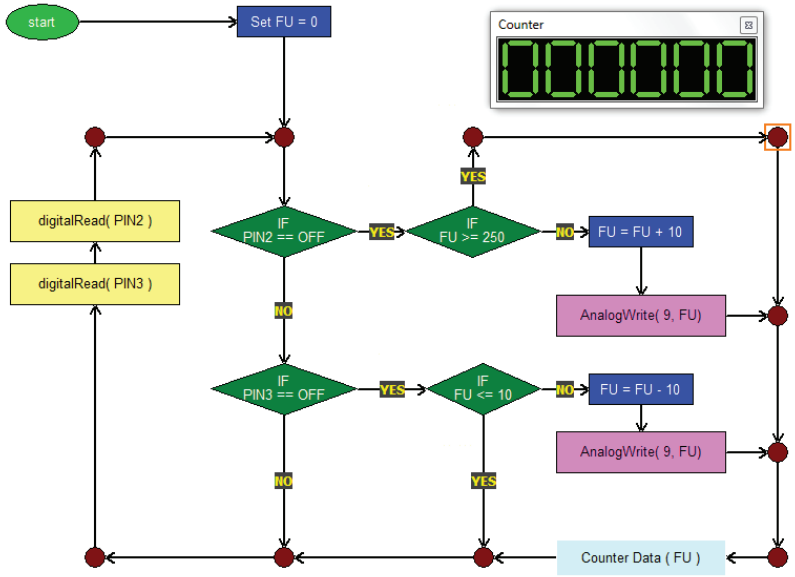
EXAMPLE 2: PUSH BUTTON ON/OFF

This Flowprogram will turn ON the LED on PIN 9 when the Start (Pin 2) button is pressed and turn it OFF when the Stop (Pin 3)button pressed.



EXAMPLE 3: PUSH BUTON FADE UP/DOWN

This Flowrogram written to Fade UP and DOWN the LED connected to Pin 9. When Push button on Pin 2 (START) is presed, the LED will become brighter and Push button on Pin 3 (STOP) is pressed, the LED will dim.



Data Block

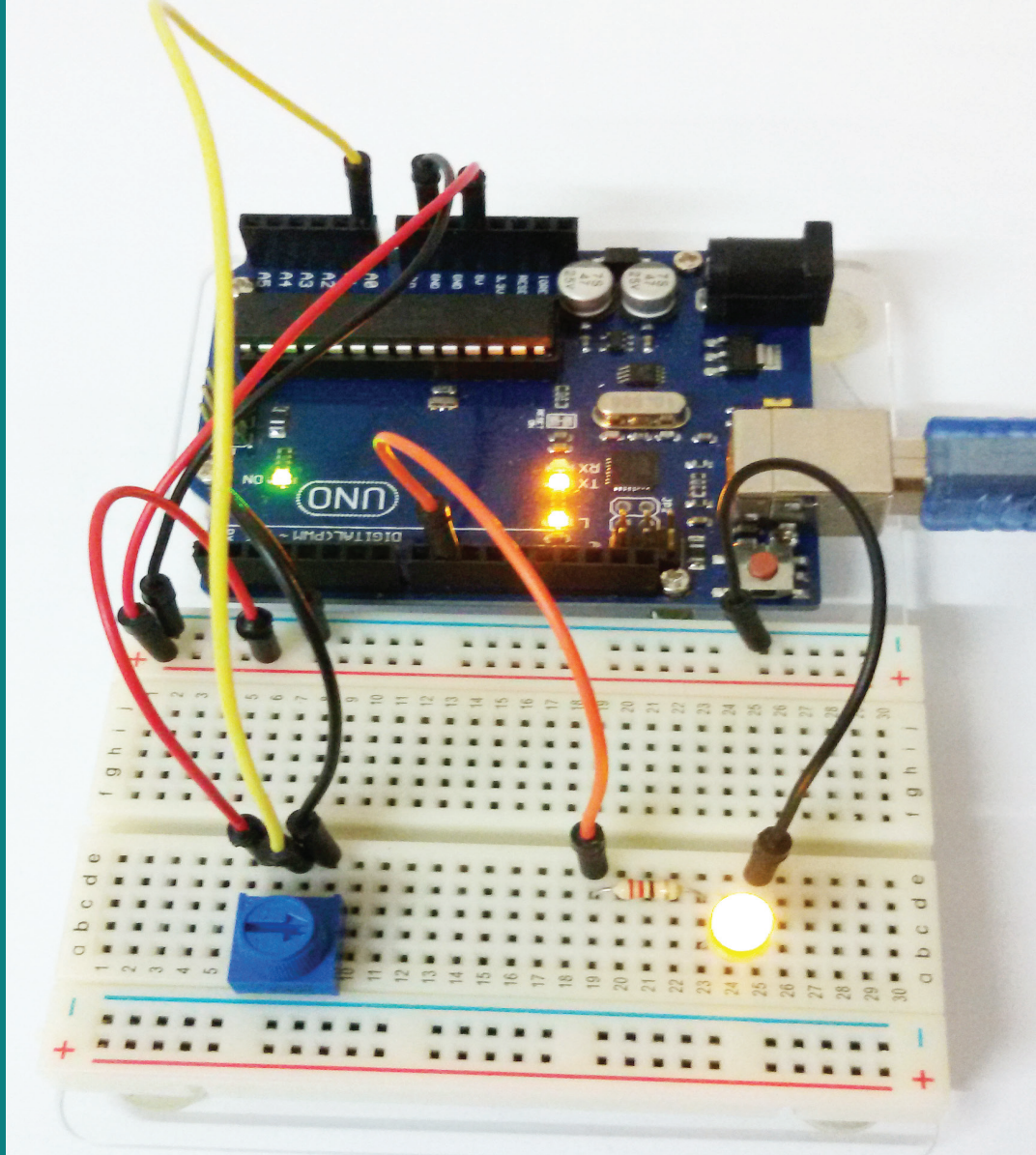
To display the value of FU on the Counter window

CHAPTER 10D

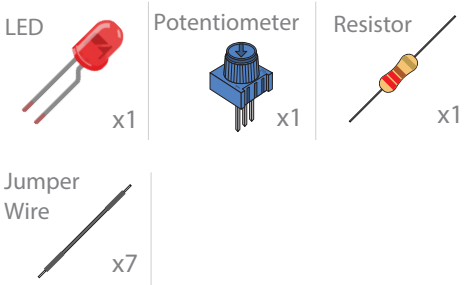
POTENTIOMETER

A potentiometer is an Analog Input device. It is also known as a variable resistor. When it's connected with 5 volts across its two outer pins, the middle pin outputs a voltage between 0 and 5 volts, depending on the position of the knob on the potentiometer.

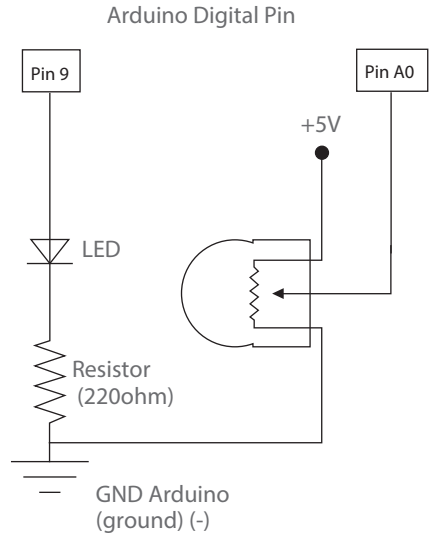
In this Experiment, you'll learn how to read the voltage produced by the potentiometer and use it to control the brightness of an LED.



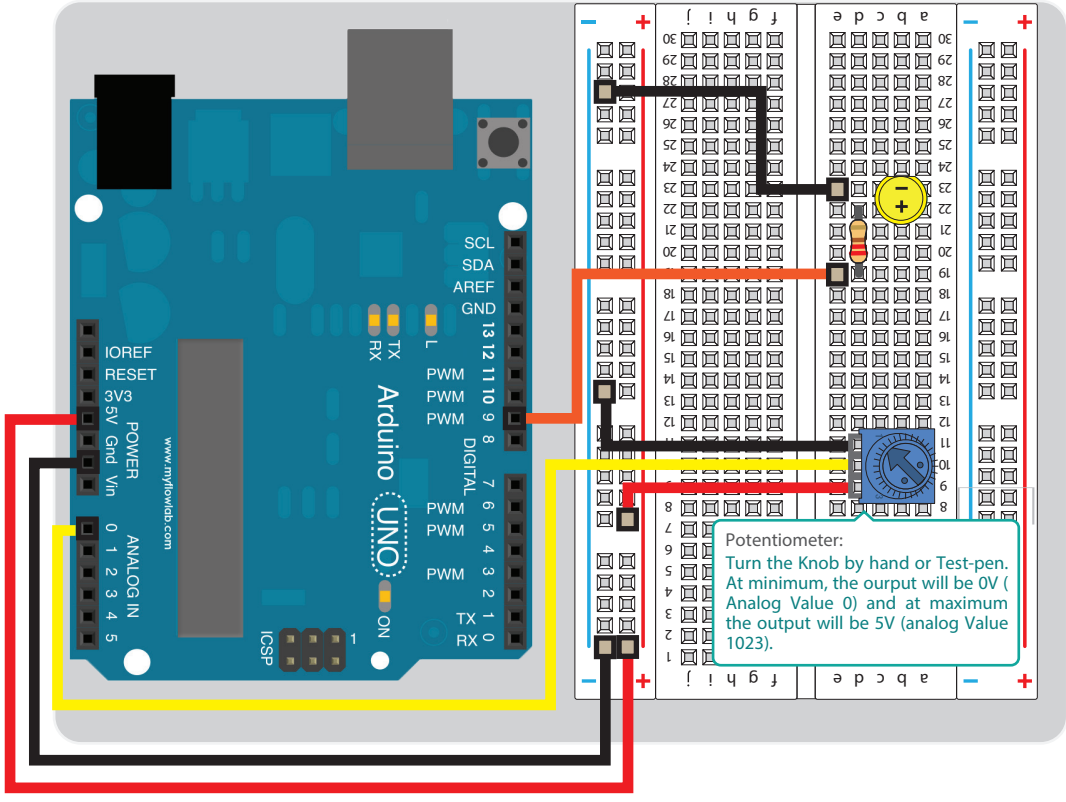
WHAT YOU WILL NEED



SCHEMATIC DIAGRAM



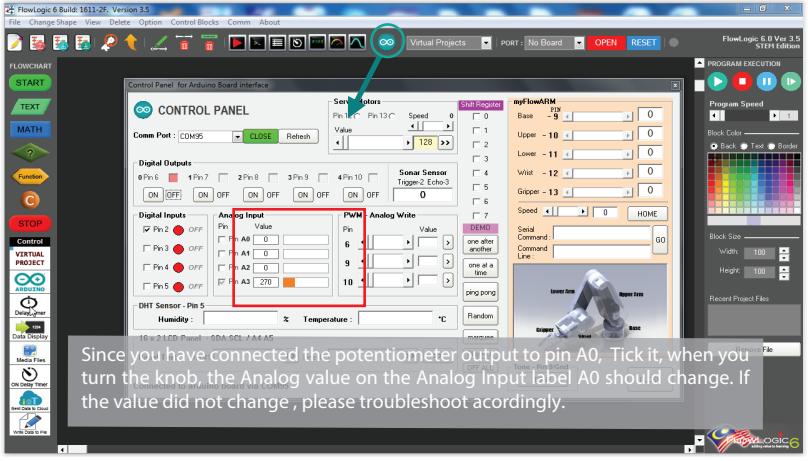
PROTOTYPING CONNECTION DIAGRAM



Disconnect Arduino Board USB connection from your PC when carrying out the wires and electronic components connection. Check thoroughly your connections before plugging in the USB cable to your PC.

TESTING YOUR CIRCUIT - POTENTIOMETER

Using the Arduino panel in Flowlogic 6 , test your Potentiometer circuit and the raw value on the connected Analog pin



NOTE : You could Enable/ disable other Analog Input Pin value display by toggling it selection box

TROUBLESHOOTING
3 THINGS TO TRY TO SOLVE THE PROBLEM

1 Working but not smooth

This is most likely due to a slightly bad connection with the potentiometer's legs. This can usually be solved by holding the Potentiometer down.

2 Not Working

Make sure you haven't accidentally connected the potentiometer's center leg to digital pin 2 rather than analog pin 2. (A0)

3 Still Not Working

If all of your connections are correct, check the potentiometer if its faulty. using multimeter check resistance between the center leg and the outer leg.

DIGITAL VERSUS ANALOG: WHAT'S THE DIFERENCE?

The devices that you have interfaced in the earlier experments, such as LEDs and push buttons, have only two possible states: ON and OFF, or "HIGH" (5 Volts) and "LOW" (0 Volts).

DIGITAL

LOW		HIGH
off	to	on
0 volts		5 volts

But there are also a lot of devices out there that are not just "ON" or "OFF". Temperature levels, control knobs, etc. all have a continuous range of values between HIGH and LOW.

ANALOG

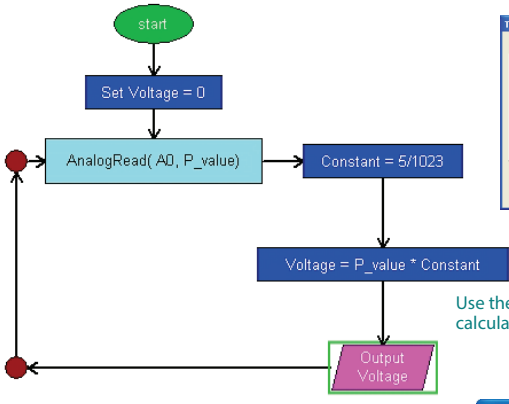
Voltage : 0 volts		Voltage : 5 volts
Input Value: 0	to	Input Value: 1023
Output Value: 0		Output Value: 255

PROGRAMMING EXAMPLES - POTENTIOMETER

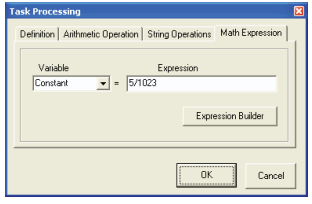
Potentiometer produces Analog value between 0 to1023 when the knob is turned clockwise or anti-clockwise., using the Analog Input from Arduino command blocks, the Analog value can be read. and convert to voltage and Analog Output value (0 -255). Based on these values you should decide the appropriate action, eg. turning on the motor with various speed or changing brightness of a LED.

EXAMPLE 1:
MEASURING POTENTIOMETER OUTPUT VOLTAGE

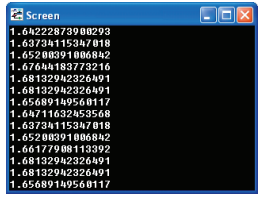
When the Flowprogram running, turn the Potentiometer Knob to change the output voltage.



Use the math Expression property to define the 'Constant' variable value.



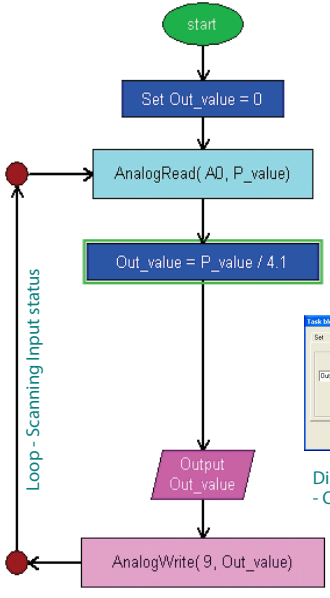
Use the Aritmatic operation property to calculate "Voltage" variable value.



Open Display console to view the voltage value.

EXAMPLE 2:
LED DIMMING USING POTENTIOMETER

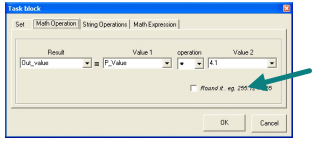
When the Flowprogram running, turn the Potentiometer Knob to Fade UP and Down the Led brigthness.



Define Variable 'Out_value' with initial value 0.

Use Analog Input block from Arduino command to read the Analog value from pin AO and store into variable 'P-value'.

Convert the 'P-Value' of (0 - 1023) to Analog Output value of (0 -255) and store it into variable 'Out_value'.



Check the Format check box to format the output value without decimal.

Display the converted value on screen .
- Open the Display console to view it realtime

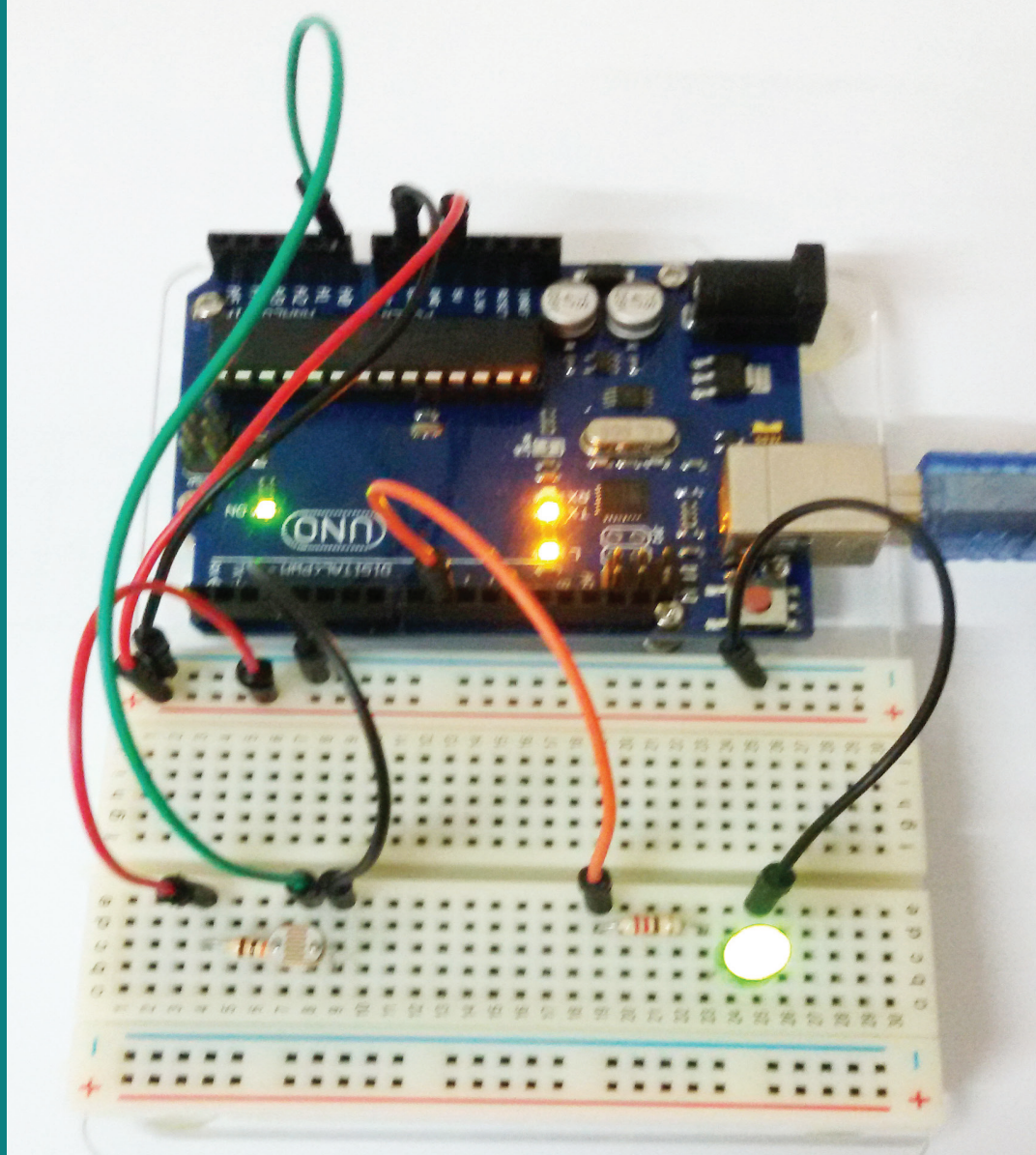
Use Analog Write from Arduino command block to write 'Out_value' variable value to pin 9.

CHAPTER 10E

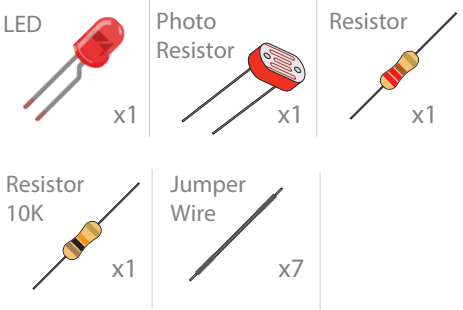
PHOTO-RESISTOR

In this circuit, you'll be using a photo resistor (light sensor), which changes resistance based on how much light the sensor receives. Arduino can't directly interpret resistance as it reads voltage, so we use a voltage divider in our circuit. The divider will output a voltage to the Analog pin. Please refer to "What is a voltage divider ?" in this chapter to learn more.

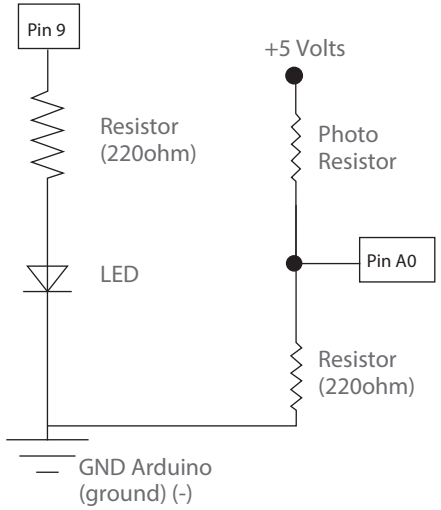
In this Experiment, you'll learn how to use a Photo Resistor resistance to control the brightness of an LED.



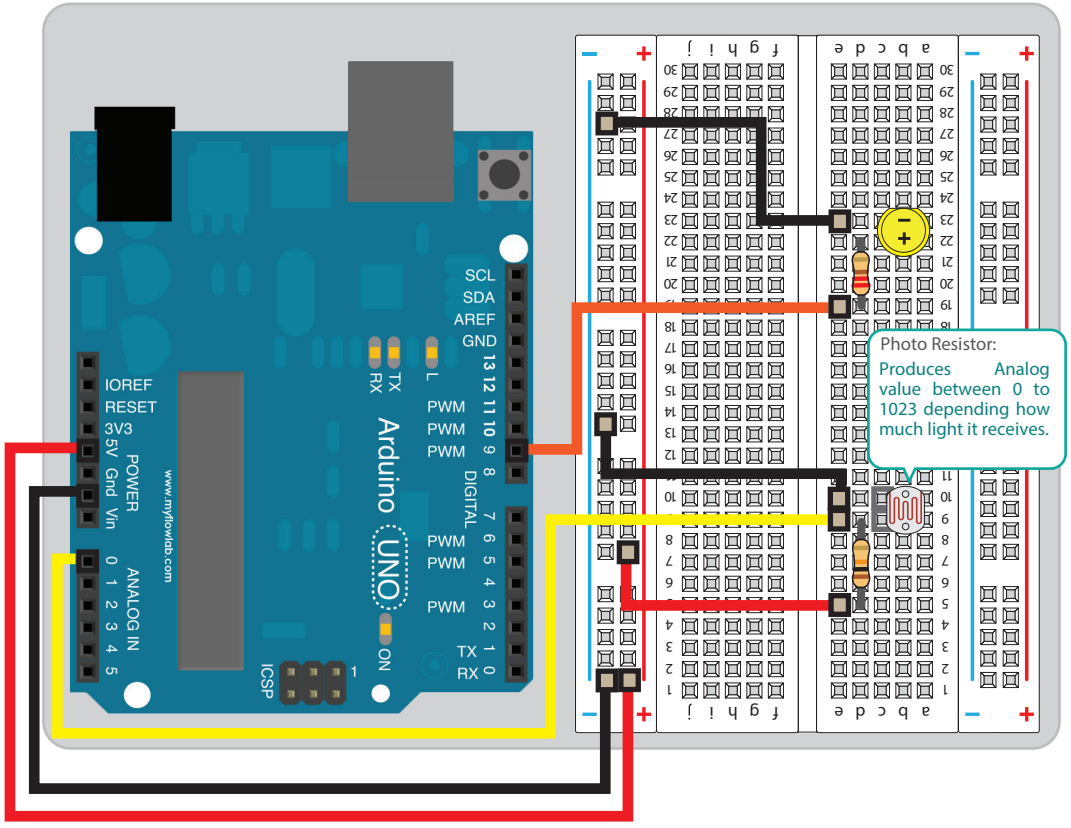
WHAT YOU WILL NEED



SCHEMATIC DIAGRAM



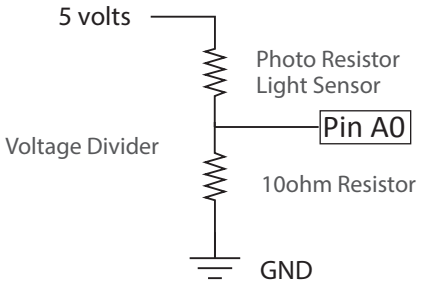
PROTOTYPING CONNECTION DIAGRAM



Disconnect Arduino Board USB connection from your PC when carrying out the wires and electronic components connection. Check thoroughly your connections before plugging in the USB cable to your PC.

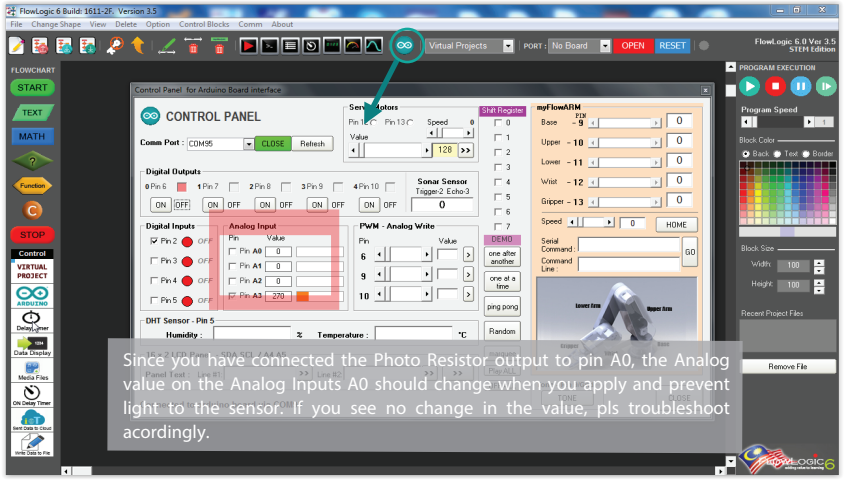
WHAT IS A VOLTAGE DIVIDER ?

A voltage divider consists of two resistors. the "top" resistor is the Photo-Resistor you'll be using. the "bottom" one is a 10 ohm resistor . When you connect the top resistor to 5 Volts, and the bottom resistor to ground, the middle will output a voltage proportional to the values of the two resistors. When one of the resistors changes (as it will when your sensor senses things), the output voltage to pin A0 will change as well.



TESTING YOUR CIRCUIT - PHOTO RESISTOR

Using the Arduino panel in Flowlogic 6 , test your Photo-Resistor circuit and the raw Analog input value on the selected pin



TROUBLESHOOTING
3 THINGS TO TRY TO SOLVE THE PROBLEM

1 **LED Remains Dark**
This is a mistake we continue to make time and time again, if only they could make a LED that works both ways. Pull it up and give it a twist.

2 **Not responding to changes in Light**
Given that the spacing of the wires on the photo-resistor is not standard, it is easy to misplace it. Double check it's in the right place.

3 **Still Not Working**
If all of your connections are correct, check the Photo Resistor if its faulty. Replace it and try again.

PROGRAMMING EXAMPLES - PHOTO-RESISTOR

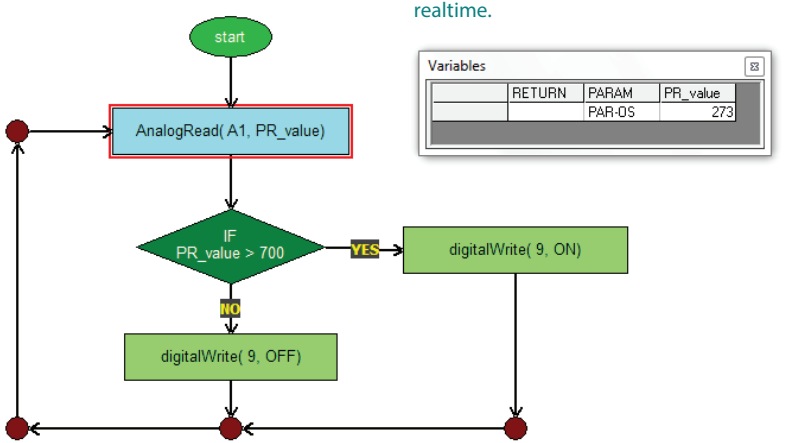
Photo Resistor produces Analog value (0 - 1023) accordance with how much light your photoresistor is reading. Using the Analog Input from Arduino command blocks the Analog value can be read.

From the Arduino control panel, you should identify what is the value produced at Pin A0 when the Photo Resistor is exposed to lot of light and when there is less light. Based on the Analog value produced you should decide the action, eg. turning on the LED with different brightness.

EXAMPLE 1: DAY NIGHT SENSING

This Flowprogram example will read the Analog value continously from the Photo Resistor connected to Pin A0, turn ON the LED (connected to Digital Pin 9) when dark and turn OFF when bright.

Click the Variable Monitor widget from the toolbar to view the variable PR_value realtime.

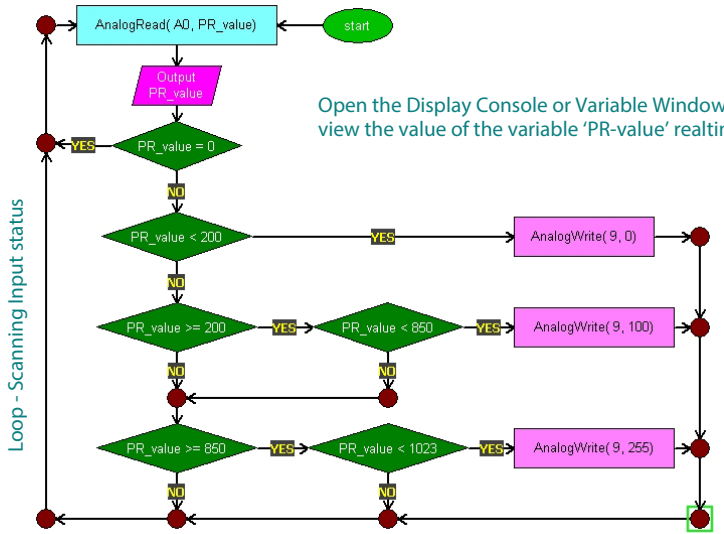


Define the Decision block property PR_Value comparison value as per your prototype

EXAMPLE 2 : LED DIMMING BASED ON AMBIENT LIGHT

This Flowprogram example will read the Analog value continously from the Photo-Resistor and turn ON the LED with different brightness using Analog Write block.

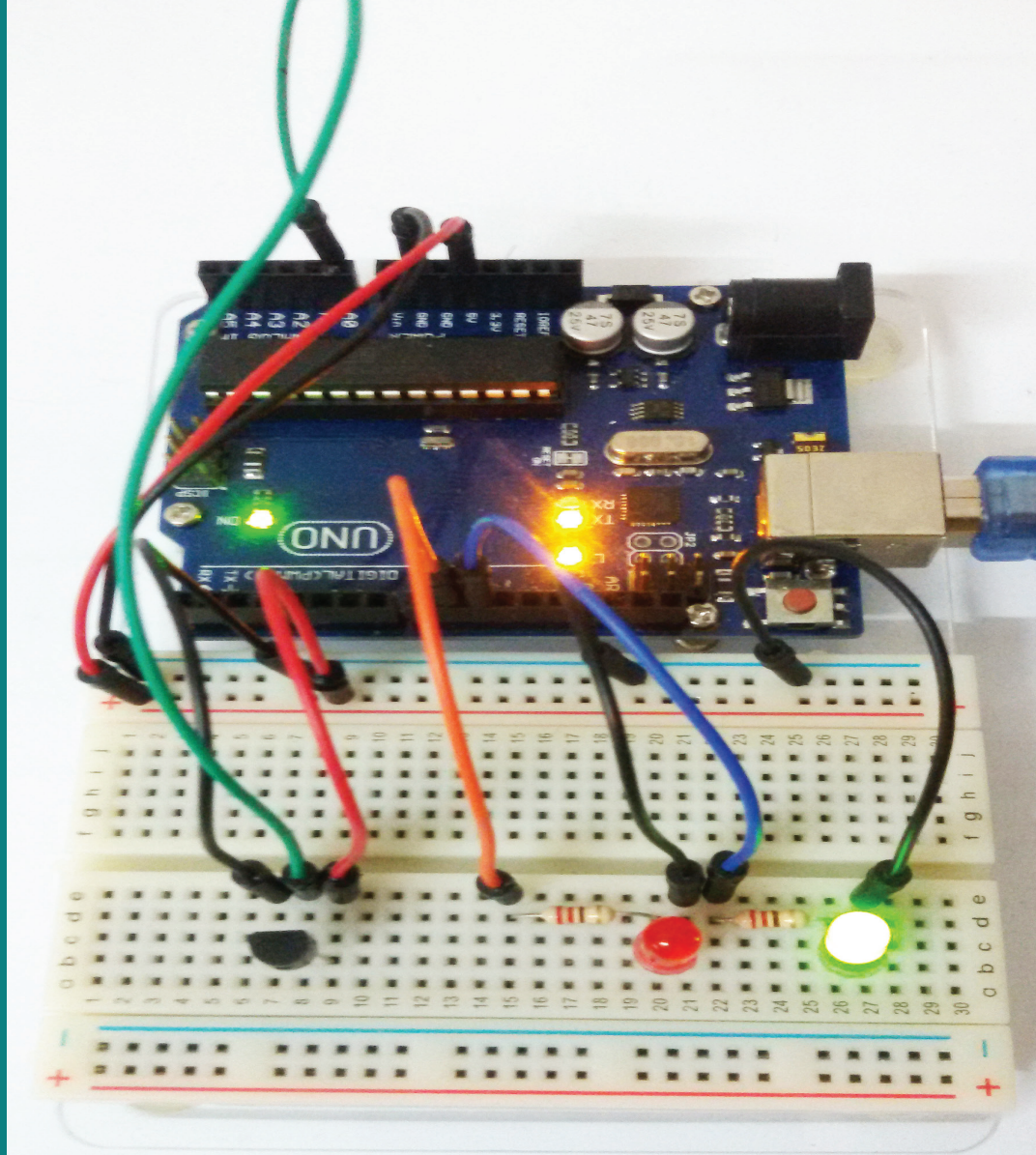
Open the Display Console or Variable Window to view the value of the variable 'PR-value' realtime.



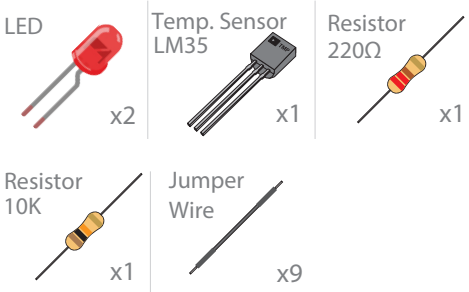
CHAPTER 10F

TEMPERATURE SENSOR - LM35

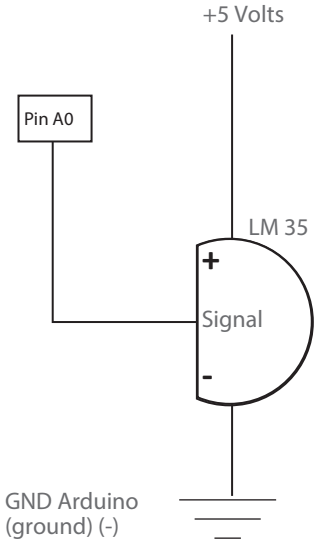
A temperature sensor is used to measure an ambient temperature. This sensor has three pins – a positive, a ground, and a signal. In this circuit, you'll learn how to Integrate the temperature sensor with your Arduino, read the value and convert to Voltage (V), Celsius (C) and Fahrenheit (F). We will also integrate LEDs to create an application to indicate temperature status.



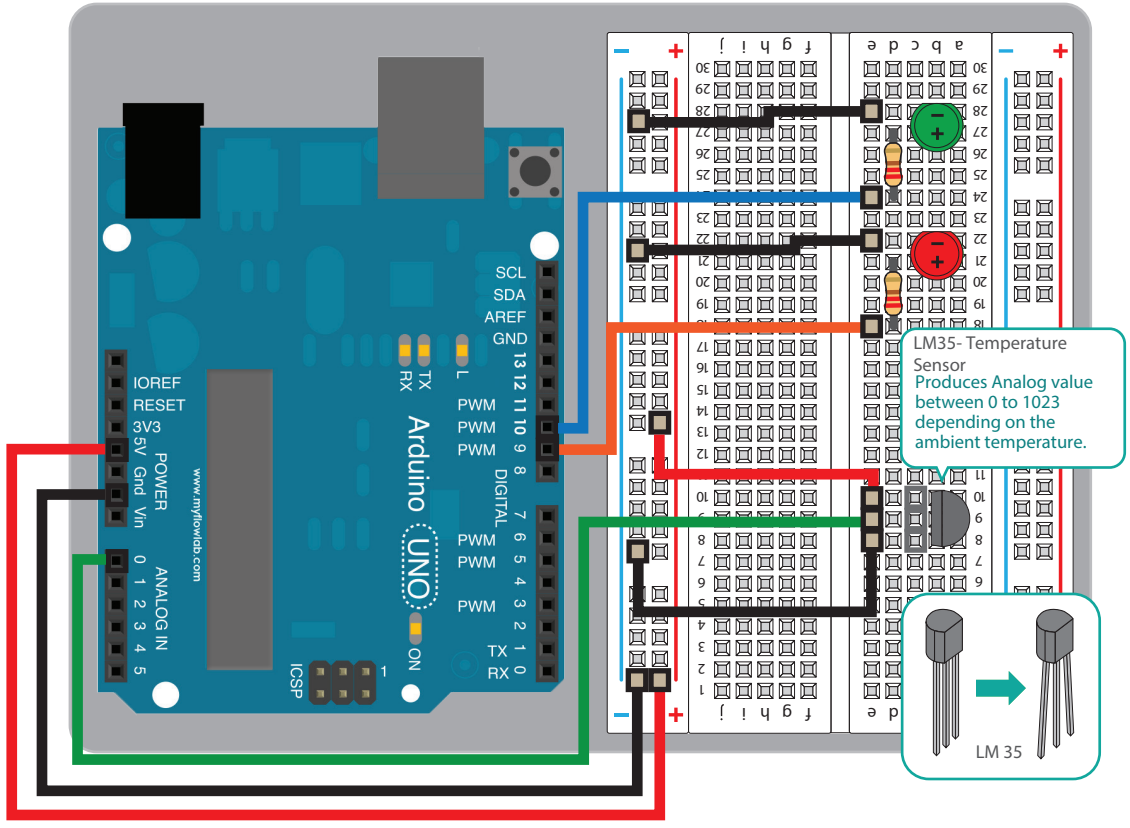
WHAT YOU WILL NEED



SCHEMATIC DIAGRAM



PROTOTYPING CONNECTION DIAGRAM



Disconnect Arduino Board USB connection from your PC when carrying out the wires and electronic components connection. Check thoroughly your connections before plugging in the USB cable to your PC.

HOW A LM35 TEMPERATURE SENSOR WORKS

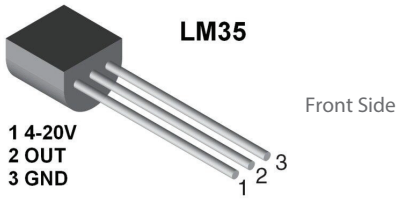
An analog temperature sensor that is easy to explain. It is a chip that tells you what the `ambient temperature is. When connected to Arduino board Analog pin, it generates an Analog value ranging from 0 -1023 proportional to temperature.

FORMULAS FOR CONVERSION

Analog value to Voltage : TVoltage = Analog Value*0.0049

Analog value to celcius : TempC = (Analog Value*500)/1024

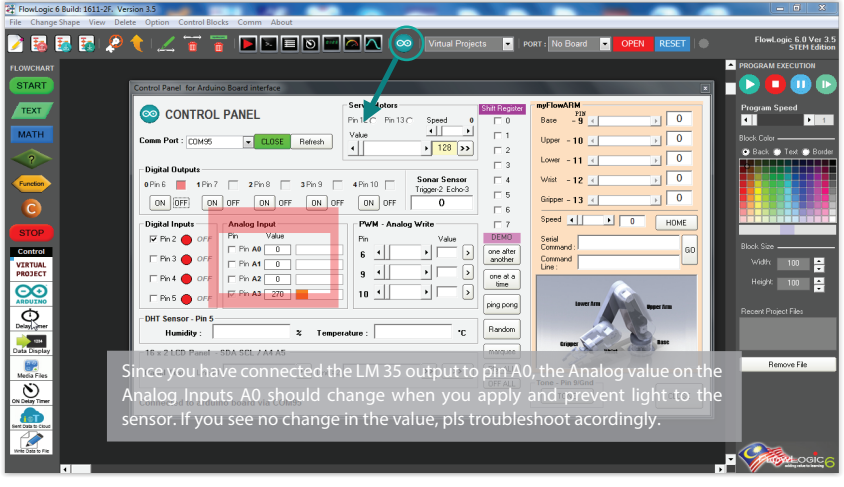
Celcius to Fahrenhiet : TempF = ((TempC*9)/5) + 32



Be careful not to mix up the LM 35 temperature sensor with other transistor, they're almost identical. Read the label on front side of the device first.

TESTING YOUR CIRCUIT - LM35 TEMPERATURE SENSOR

Using the Arduino panel in FlowLogic 6 , test your LM 35 Temperature sensor circuit and the raw value on the Analog pin it is connected.



TROUBLESHOOTING
3 THINGS TO TRY TO SOLVE THE PROBLEM

1

Nothing Seems to Happen

If you running Example 1, the Flowprogram has no indication it is working. To view the results you must open the variable window from the view menu.

2

Temperature Value is Unchanging

Try pinching the sensor with your fingers to heat it up or pressing a bag of ice against it to cool it down.

3

Still Not Working

If all of your connection are correct, check the LM35 Temperature sensor, if its faulty. Replace it and try again.

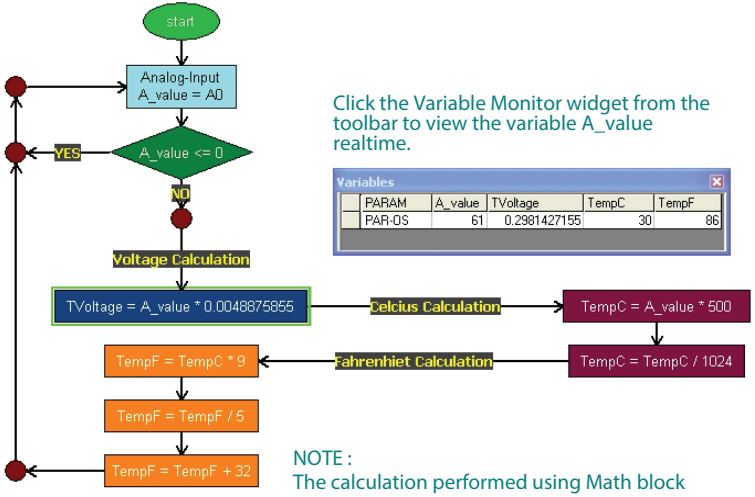
PROGRAMMING EXAMPLES - LM35 TEMPERATURE SENSOR

LM35 Temperature Sensor produces Analog value between 0 to 1023 in accordance with the ambient temperature. Using the Analog Input from Arduino command blocks the Analog value can be read.

From the Arduino control panel, you should identify what is the value produced at Pin A0 when the LM35 temperature sensor exposed to hot and cold . Use Analog value produced you should decide the action required.

EXAMPLE 1: CALCULATION

This Flowprogram example will read the Analog value continously from the LM35 Temperature sensor connected to Pin A0 and perform calculation to determine the Voltage, Temperature reading in Celcius and Fahrenhiet.



Click the Variable Monitor widget from the toolbar to view the variable A_value realtime.

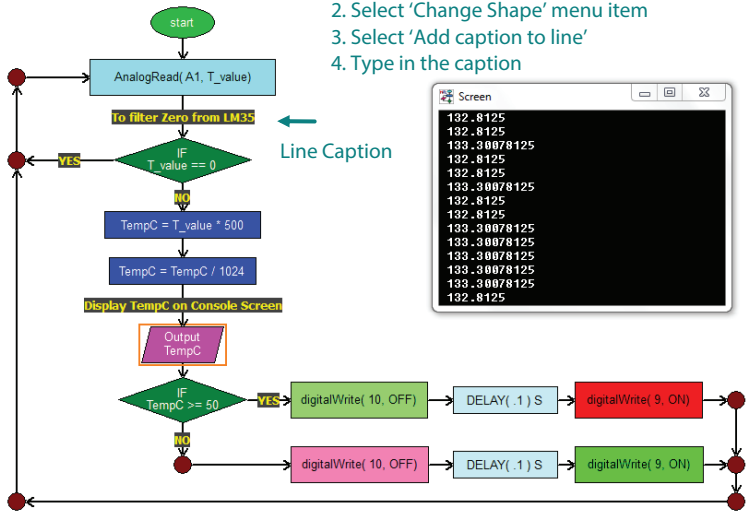
NOTE :
The calculation performed using Math block (Arithmetic + - x /) blocks in stages.

EXAMPLE 2: INDICATOR

This Flowprogram example will turn On the Red LED when Temperature is High and Green LED when temperature is at ambient reading.

To add caption to the line folow the steps below:

1. select the block after the line to place the caption
2. Select 'Change Shape' menu item
3. Select 'Add caption to line'
4. Type in the caption

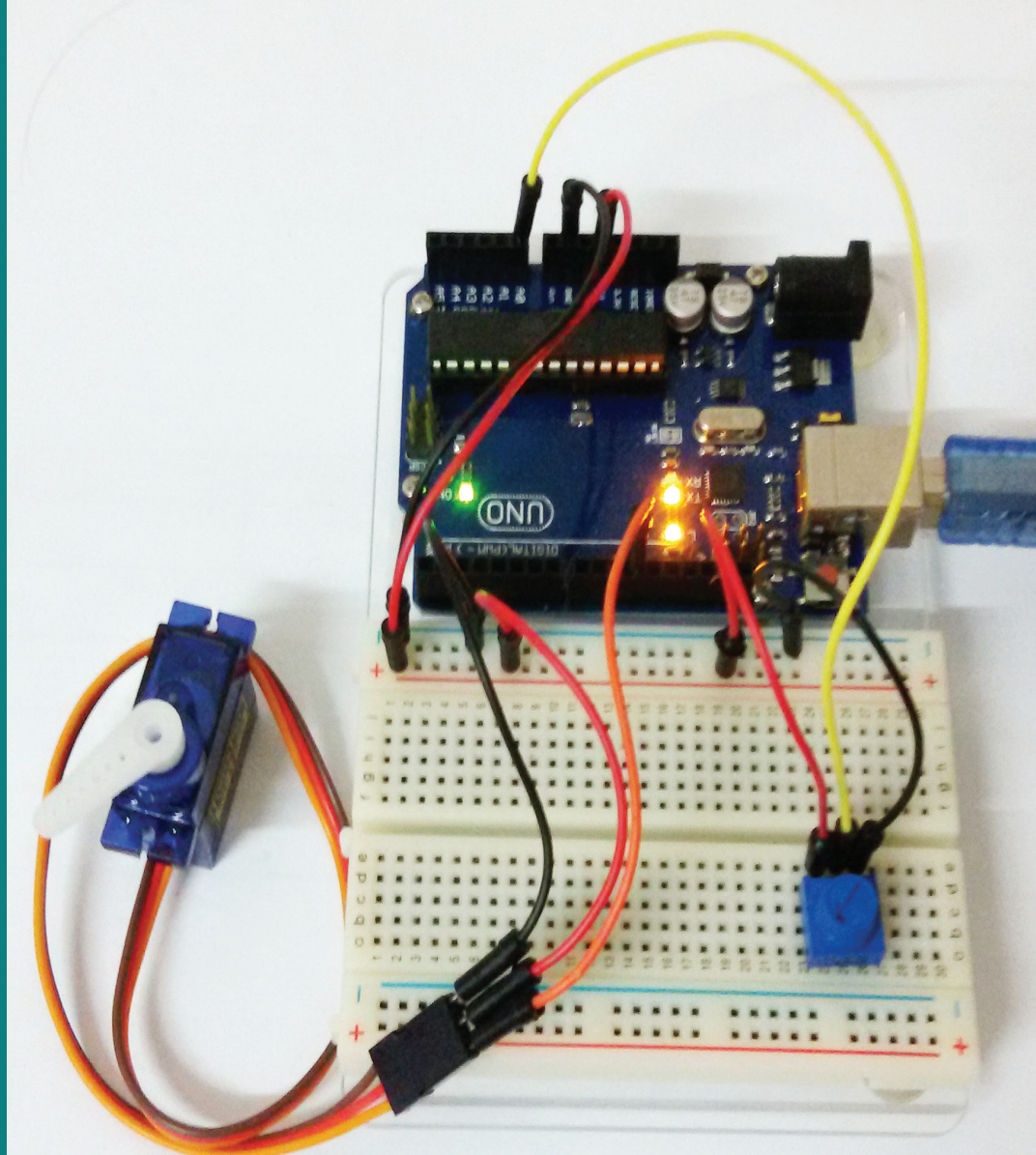


CHAPTER 10G

RC SERVO MOTOR

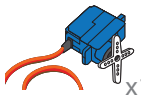
RC Servo motors are ideal for motion applications that requires positioning. In this experiment, you'll learn how to use Servo command blocks to activate and position the shaft at various angles.

We will also mount a potentiometer as a knob to position the Servo motor. Using FlowLogic 6 we will develop Flowpogram to integrate this device to control the RC Servo motor.



WHAT YOU WILL NEED

RC Servo motor

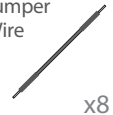


x1

Potentiometer

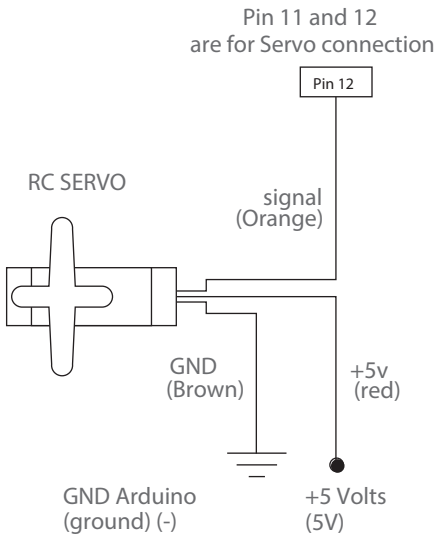
x1

Jumper Wire

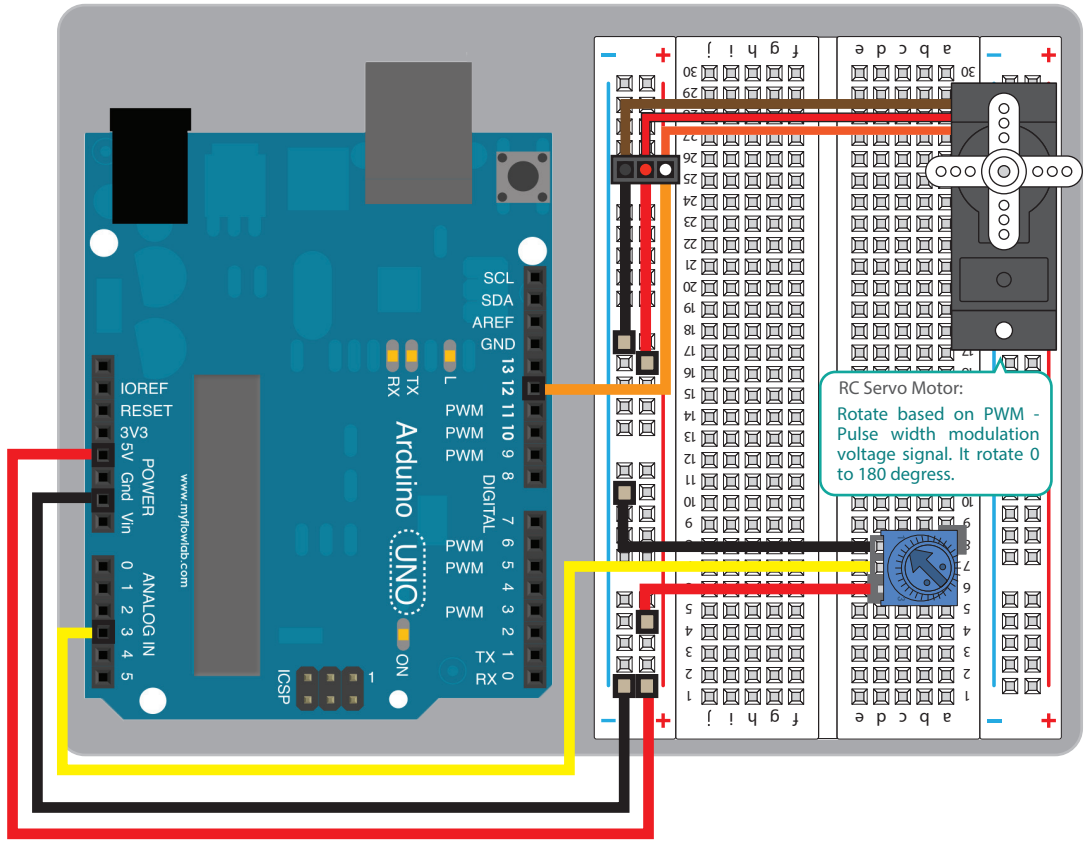


x8

SCHEMATIC DIAGRAM



PROTOTYPING CONNECTION DIAGRAM

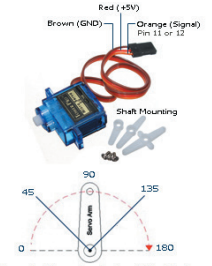


Disconnect Arduino Board USB connection from your PC when carrying out the wires and electronic components connection. Check thoroughly your connections before plugging in the USB cable to your PC.

HOW A RC SERVO MOTOR WORKS

A servomotor is a rotary actuator that allows for precise control of angular position. The angle is determined by the duration of a pulse that is applied to the control wire. This is called Pulse width Modulation (PWM) . The servo expects to see a pulse every 20 ms. The length of the pulse will determine how far the motor turns. For example, a 1.5 ms pulse will make the motor turn to the 90 degree position (neutral position). Normally a RC servo motor rotate between 0 to 180 degree.

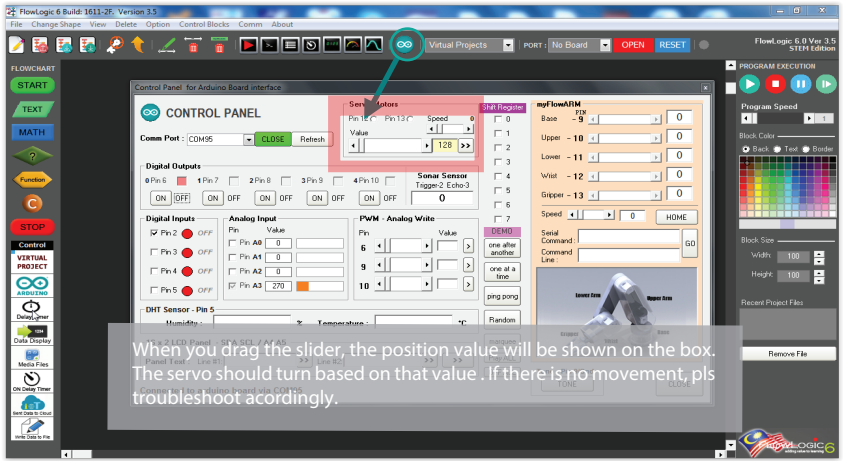
Using FlowLogic 6 you can position the motor shaft easily by sending the angle value between 0 to 180 using the Servo Control block from the Arduino commad.



Servo Position using Analog Write block from Arduino command block

TESTING YOUR CIRCUIT - RC SERVO MOTOR

The Arduino board provides Pin 11and Pin 12 for Servo motor connection. Select the pin option where you servo motor is connected and drag the slider to test it, the value is 0 to 180 degree.



TROUBLESHOOTING
3 THINGS TO TRY TO SOLVE THE PROBLEM

1 Servo Not Twisting

Even with colored wires it is still shockingly easy to plug a servo in backward.This might be the case.

2 Still Not Working

A mistake we made a time or two was simply forgetting to connect the power (red and brown wires) to +5 volts and ground.

3 Fits and Starts

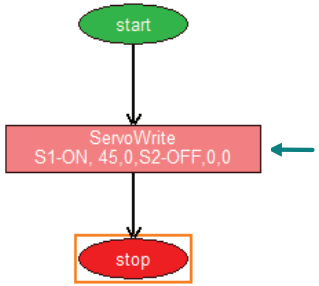
When servo twitches, and LED on the Arduino board flashing , the power supply you are using is not quite enough. Using a wall adapter instead of USB should solve it.

PROGRAMMING EXAMPLES - RC SERVO MOTOR

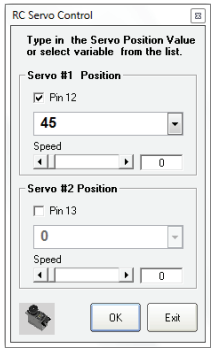
RC Servo motor requires position value in degree (between 0 to 180) to rotate to the desired angle that you wanted.

Using the Servo Control block from Arduino command, the position value can be sent to the RC servo but before you could do that, you need to use Servo setup block to Attached or Detach the servo motor.

EXAMPLE 1:
ROTATE TO AN ANGLE

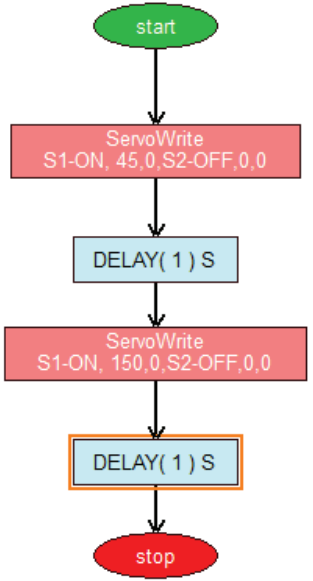


This ServoWrite Command block will move the Servo motor connected to Pin 12 to 45 Degree



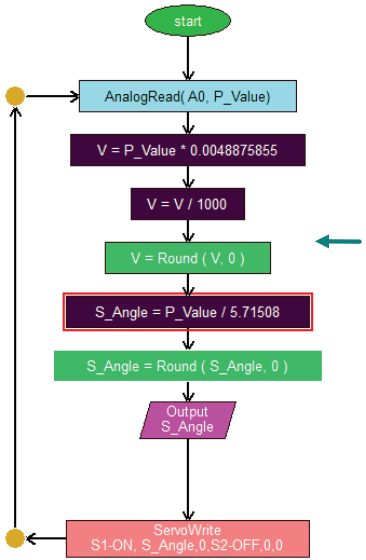
EXAMPLE 2:
CW/CCW ROTATION

Clockwise (CW)
Counter clockwise (CCW)

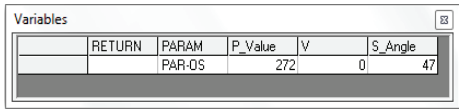


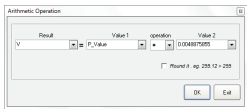
EXAMPLE 3:
POSITIONING USING POTENTIOMETER

This Flowprogram example will rotate the servo motor based on the Analog value (0 to 1023) of the Potentiometer. The Analog value will be converted to Servo Angle (0 to 180 Degree) before positioning the RC Servo motor.

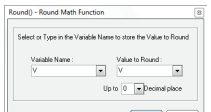


View the the variable value using Variable window

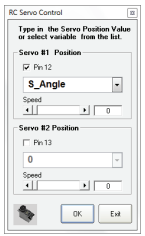




Math Block - Arithmetic Block



Math Block - Round()



ServoWrite Block -Select the S_Angle variable from the lst

CHAPTER 10H

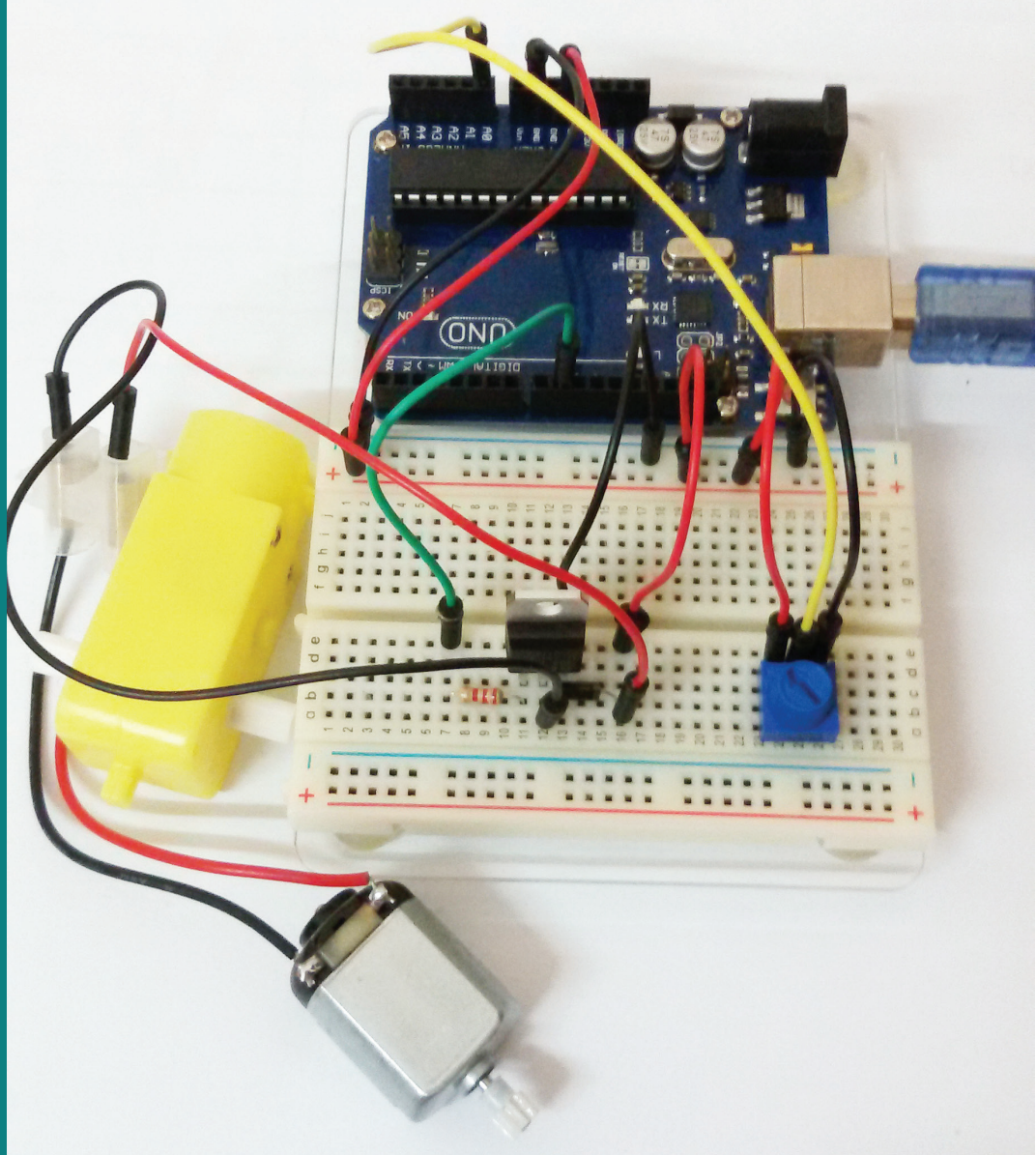
DC MOTOR

In the earlier experiment you played around with a servo motor, now we are going to tackle spinning DC motor. This requires the use of a transistor, which can switch a larger amount of current than the Arduino can handle.

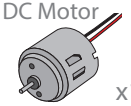
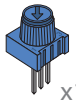
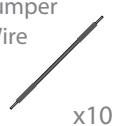
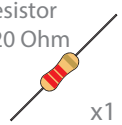
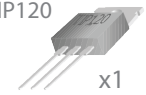
When using a transistor, you just need to make sure its maximum specs are high enough for your use.The transistor we are using for this circuit is Darlington Transistor TIP120, rated at 60V max and 5A max–more then enough for our toy motor!

We will also use Diode IN4002 to prevent reserve flow of current to protect our Arduino board and our PC.

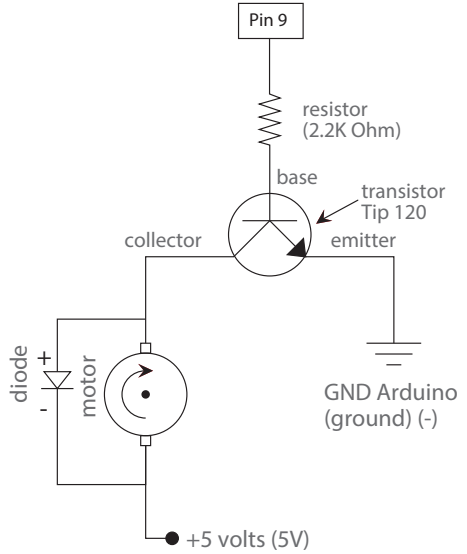
A potentiometer is also added to the circuit. Using FlowLogic 6 we will be creating Flowprogram to controll the speed of the motor.



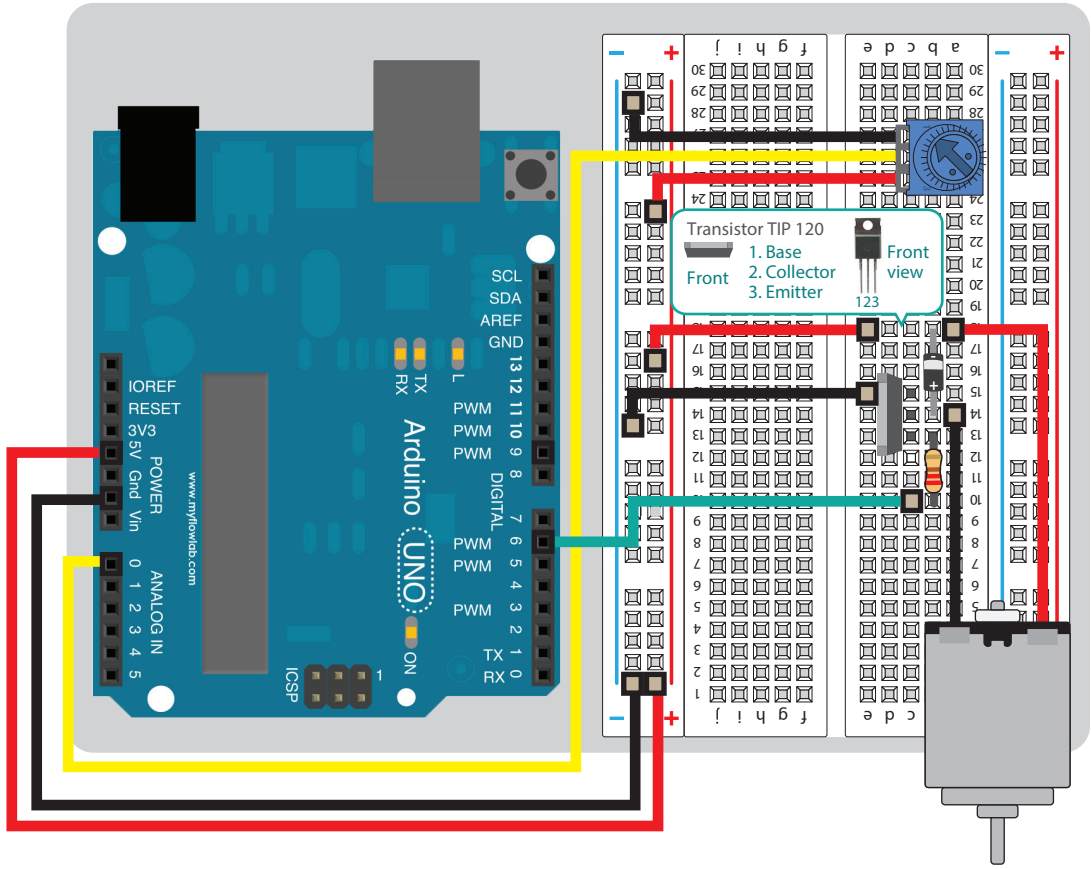
WHAT YOU WILL NEED

5 Volt DC Motor  x1	Potentiometer  x1	Jumper Wire  x10
Resistor 220 Ohm  x1	Diode IN4002  x2	Transistor TIP120  x1

SCHEMATIC DIAGRAM



PROTOTYPING CONNECTION DIAGRAM

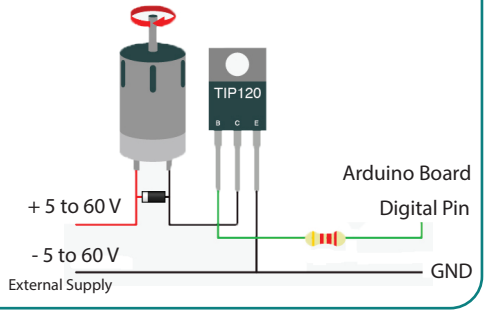


Disconnect Arduino Board USB connection from your PC when carrying out the wires and electronic components connection. Check thoroughly your connections before plugging in the USB cable to your PC.

HOW TO CONTROL A DC MOTOR

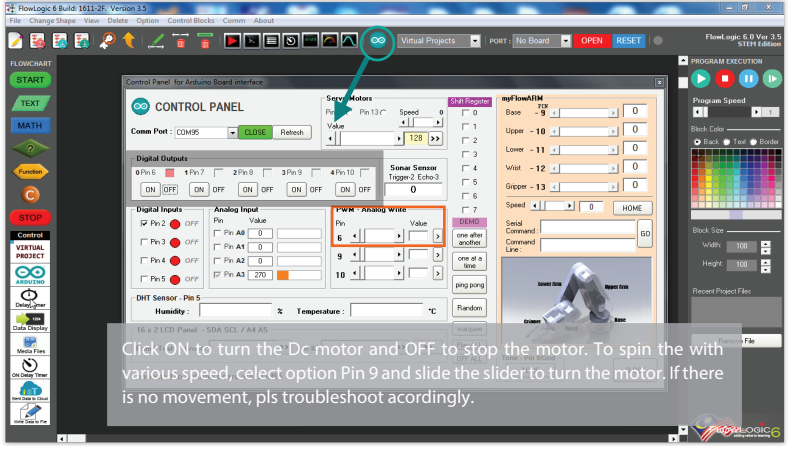
The DC motor in your Inventor kit comes with a gear head (Yellow component) . When current is PASSED through, it spins continuously in one direction until the current stops. DC motors have no polarity, meaning that you can swap the two wires over to reverse the direction of the motor. You will require a Transistor, TIP 120 and a diode to power the motor.

If you place the diode the wrong way, the current bypasses the motor and you create a short circuit. The short circuit tries to ground all the available current and could break your USB port due to over drawing too much currect.



TESTING YOUR CIRCUIT - DC MOTOR

You could connect the DC motor output to any of the Arduino output pin, but to control the speed of motor you only can connect to Digital Pin 6 and 9 (PWM). In our experiment, we have connected to Pin 9.



TROUBLESHOOTING
3 THINGS TO TRY TO SOLVE THE PROBLEM

1 **Motor Not Spinning**
If you sourced your own transistor, double check with the data sheet that the pinout is compatible with a TIP 120 and connection is correct.

2 **Still Not Working**
If you sourced your own motor, double check that it will work with 5 volts and that it does not draw too much power.

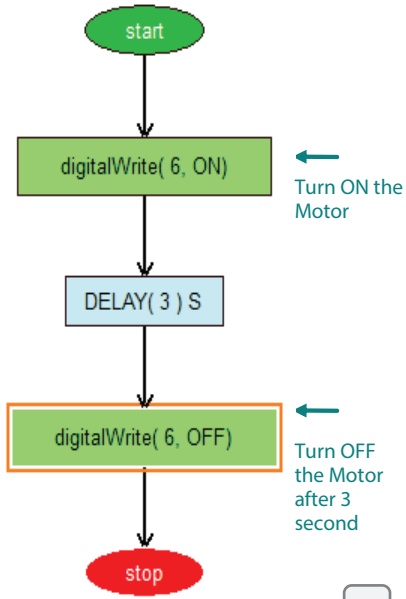
3 **Still Not Working**
Sometimes the Arduino board will disconnect from the computer. Try un-plugging and then re-plugging it into your USB port.

PROGRAMMING EXAMPLES - DC MOTOR

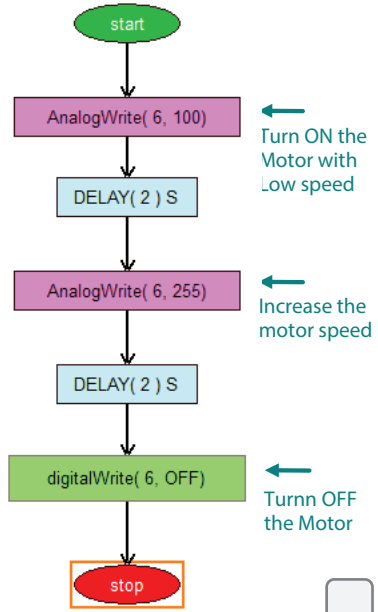
Using Flowlogic 6 you could create Flowprogram to control the DC motor. Since you have connected the motor output to digital pin 9, you can use Digital Output command from Arduino blocks to turn ON and OFF the motor. If you wish to control the motor with various speed, you should use Anlog Write command. The Analog output value is between 0 to 255.

To create Reverse/Forward (REV/FWD) application, you will need additional circuit. Please refer to our online learning resources.

EXAMPLE 1:
FREE SPINING

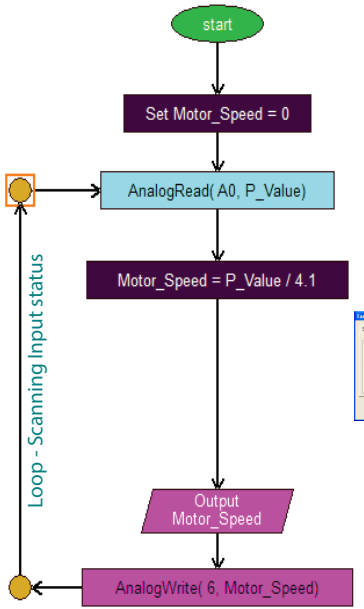


EXAMPLE 2:
CHANGE SPEED



EXAMPLE 3:
CHANGING SPEED USING POTENTIOMETER

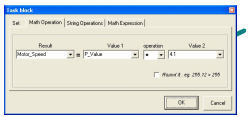
When the Flowprogram running, turn the Potentiometer Knob to change the motor speed.



Define Variable 'Motor_speed' with initial value 0.

Use Analog Input block from Arduino command to read the Analog value from pin AO and store into variable 'P-value'.

Convert the 'P-Value' of (0 - 1023) to Analog Output value of (0 -255) and store it into variable 'Motor_Speed'.



Check the Format check box to format the output value without decimal

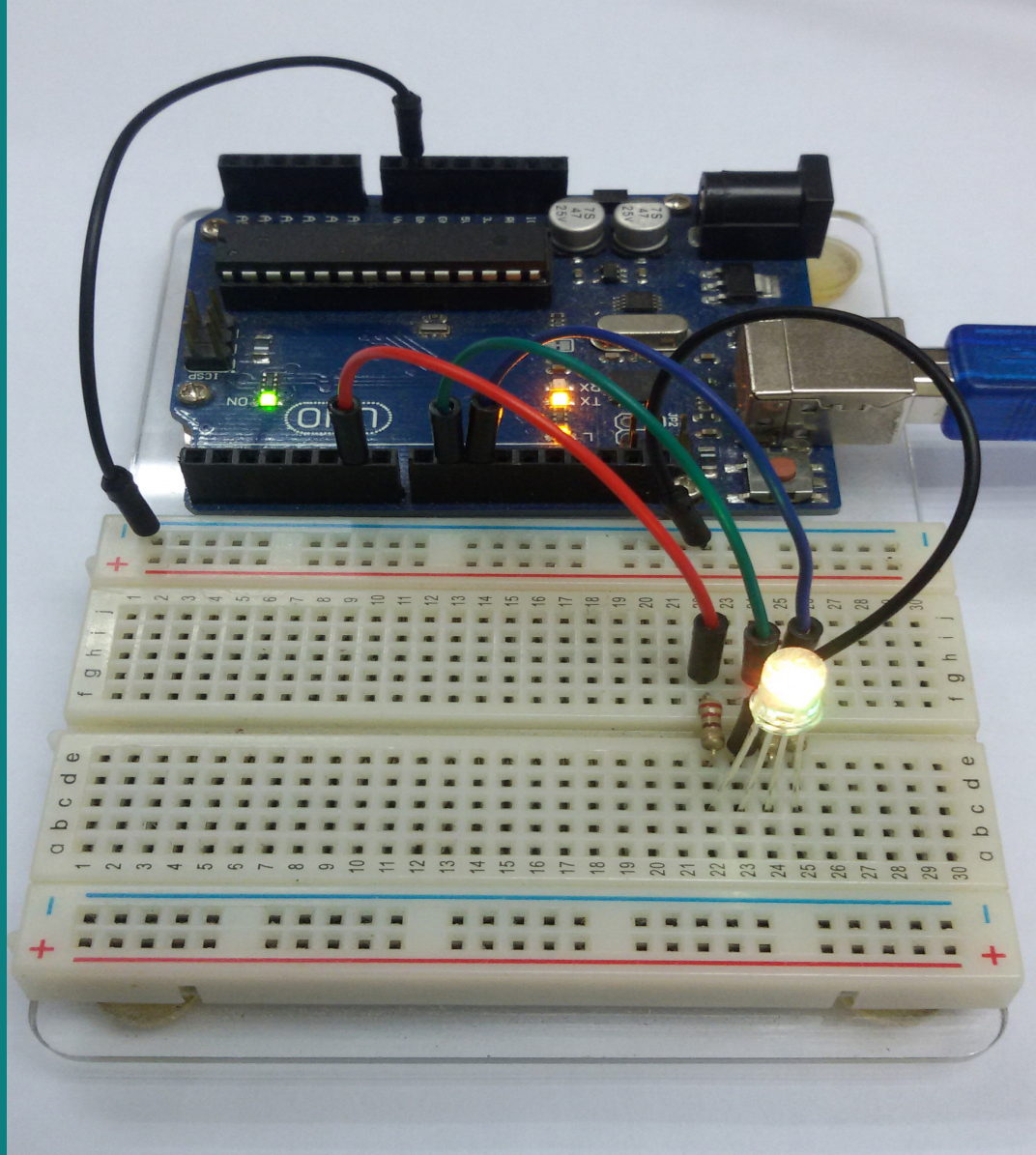
Display the converted value on screen.
- Open the Display console to view it realtime

Use Analog Write from Arduino command block to write 'Motor_speed' variable value to pin 6.

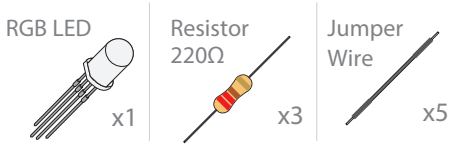
CHAPTER 10G

RBG LED

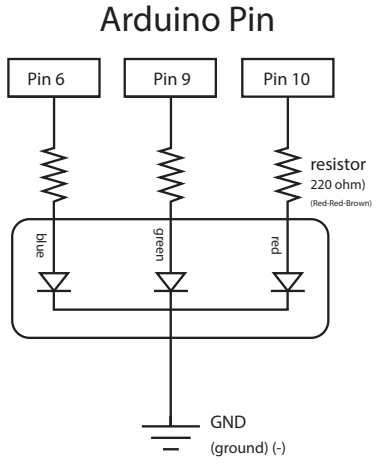
You know what's even more fun than a blinking LED? A colored one. RGB, or red-green-blue, LEDs have three different color-emitting diodes that can be combined to create all sorts of colors. In this circuit, you'll learn how to use an RGB LED to create unique color combinations. Depending on how bright each diode is, nearly any color is possible!



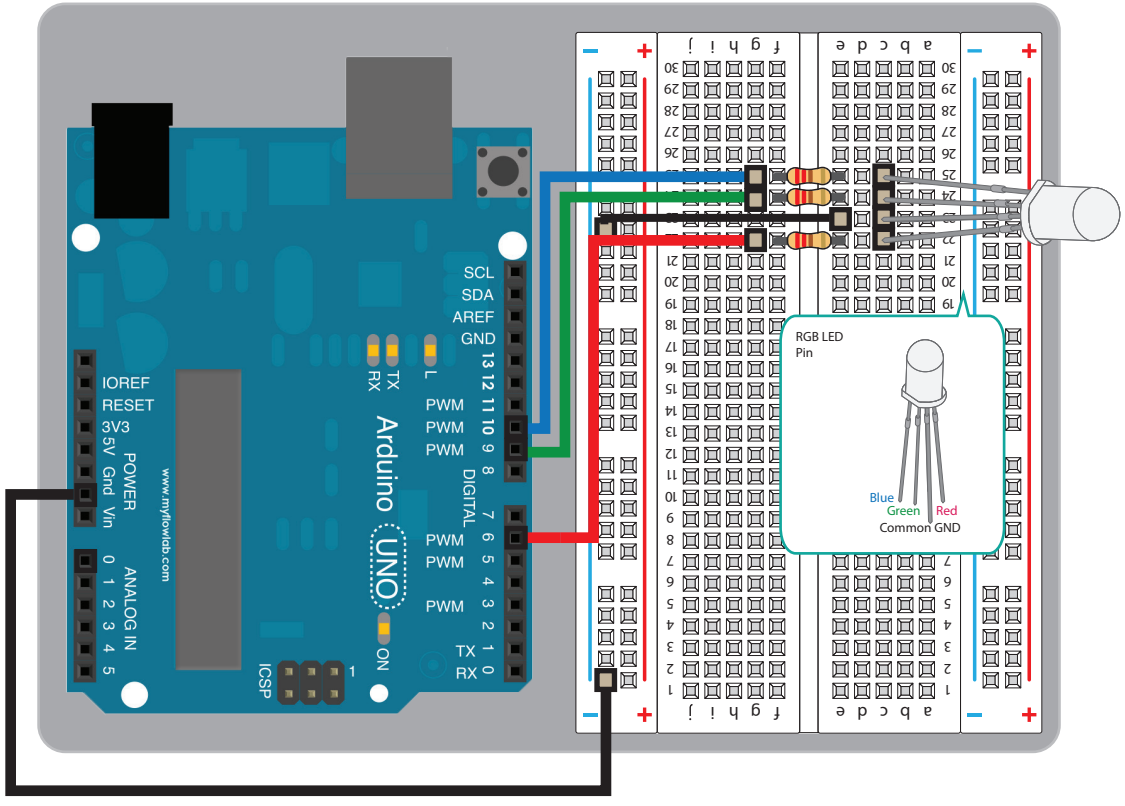
WHAT YOU WILL NEED



SCHEMATIC DIAGRAM



RGB LEDCONNECTION DIAGRAM



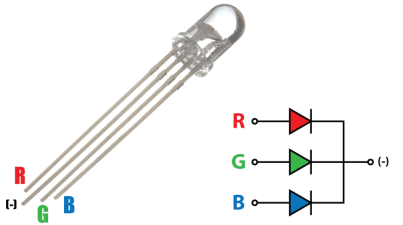
Disconnect Arduino Board USB connection from your PC when carrying out the wires and electronic components connection. Check thoroughly your connections before plugging in the USB cable to your PC.

HOW A RGB LED WORKS

At first glance, RGB (Red, Green, Blue) LEDs look just like regular LEDs, however, inside the usual LED package, there are actually three LEDs, Red, Green and Blue. By varying the brightness of each of the individual LEDs you can mix pretty much any color you want. We mix colors just like you would mix audio with a 'mixing board'.

Using Flowlogic 6 Arduino Control panel, you could mix colors to produce any color by adjusting the brightness of each of the three LEDs by scrolling the PWM Analogwrite scroll bar.

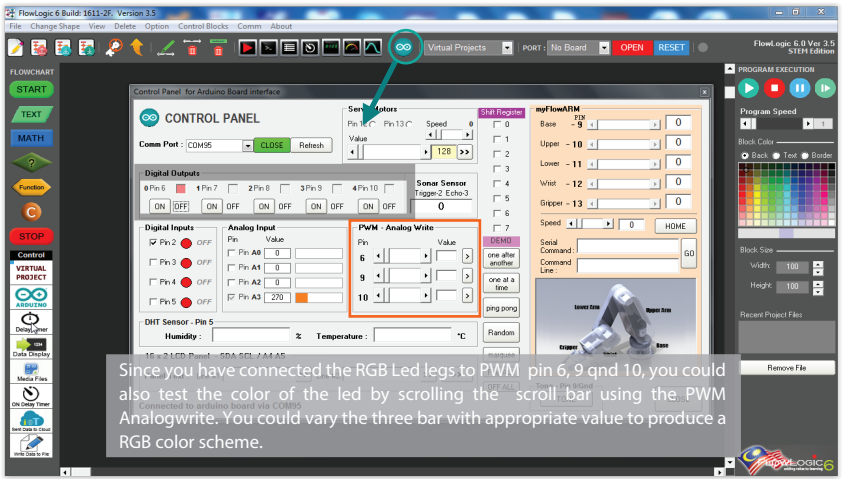
Please ensure you have connected RED LEDs leg to Pin 6, GREEN Color to Pin 9 and BLUE color to Pin 10. These Pins produces PWM signal that you can vary the voltage to the LEDs.



Use to the COMMON CATHODE RGB LED, which the long should be connected to GND Pin.

TESTING YOUR CIRCUIT - RGB LED

Using the Arduino panel in Flowlogic 6 , test your RGB LED circuit by turn ON and OFF on the Digital output pin 6,9,10. Double check each pin where it is connected and fuctioning proferly.



TROUBLESHOOTING
3 THINGS TO TRY TO SOLVE THE PROBLEM

1

Nothing Seems to Happen

If you running Example 1, the Flowprogram has no indication it is working. To view the results you must open the variable window from the view menu.

2

LED Remains Dark or Shows Incorrect Color

With the four pins of the LED so close together, it's sometimes easy to misplace one. Double check each pin is where it should be.

3

Still Not Working

If all of your connection are correct, check the RGB LED, if its faulty. Replace it and try again.

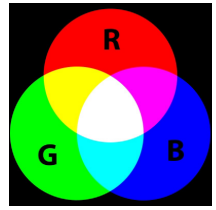
PROGRAMMING EXAMPLES - RGB LEDS

If we turn ON using DigitalWrite command block all the Pin the LEds are connected, then the overall color of the light will be white. If we turn OFF the blue LED, so that just the red and green LEDs are the same brightness, then the light will appear yellow.

We can control the brightness of each of the red, green and blue parts of the LED separately, making it possible to mix any color we like by using AnalogWrite command block.

Example RGB brightness value and color using Analogwrite

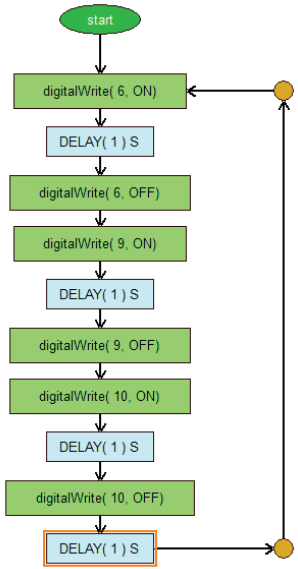
RED	GREEN	BLUE	COLOR
255	0	255	Purple
0	49	206	Yellow
250	59	43	Cyan
255	0	0	RED
0	255	0	Green
0	0	- 255	Blue

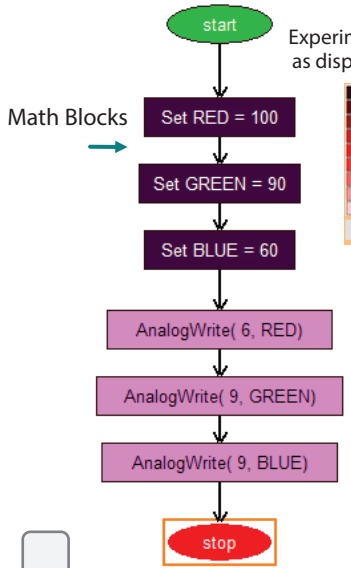


EXAMPLE 1: TURN ON AND OFF RGB LEDS

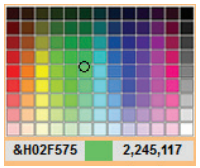
This Flowprogram example will turn ON and OFF each Led individually using DigitalWrite and AnalogWrite command block.

The AnalogWrite value can be between 0 to 255. Experiment it by varying the value. of variable RED, GREEN and BLUE.





Experiment with the RGB value as displayed on the color chart



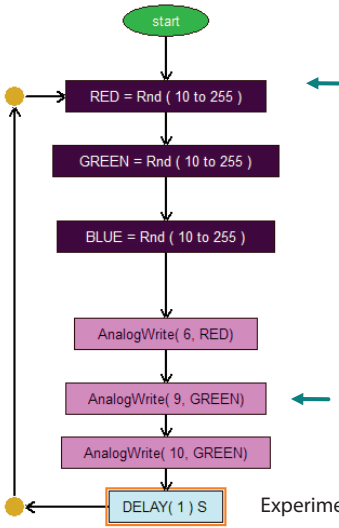
RGB Code

Click on th color to view RGB code

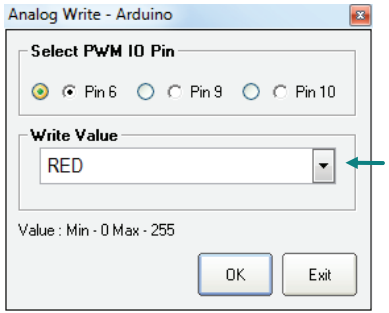
R = RED
G = GREEN
B = BLUE

EXAMPLE 2: RGB COLOR MIX USING MATH RANDOM FUNCTION

This a simple Flowprogram example shows how to vary the LEDs AnalogWrite property, the brightness value by using variable. Instead of applying fix 0-255 value, you could define a variable as shown (RED,GREEN.BLUE) and vary it via simple computation.



Math Block - Use Rnd()



Select the Variable from the list

Experiment by varying the Delay value

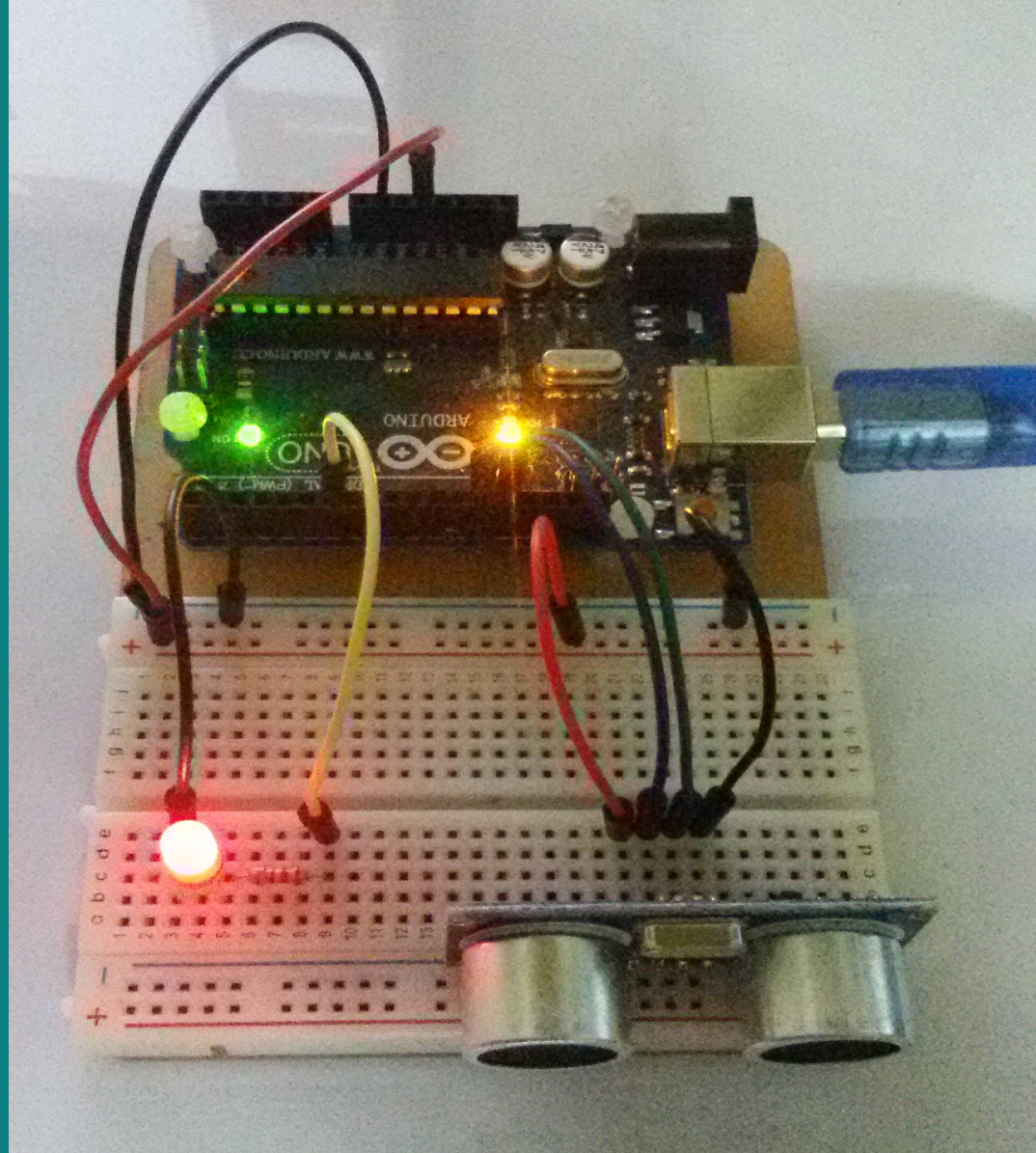
CHAPTER 10H

ULTRASONIC SENSOR

Ultrasonic sensor are great for all kind of projects that need distance measurements, avoiding obstacles as examples. It is also callef SONAR Sensor.

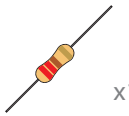
Wide range of application that requires distance sensing such example as for mobile robot, cars can be incorporated and program it for Collision Avoidance.

It essentially gives your Arduino eyes / spacial awareness and can prevent your robot from crashing or falling off a table. It has also been used in water level sensing, and even as a parking sensor



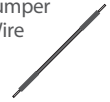
WHAT YOU WILL NEED

Resistor x1



LED x1

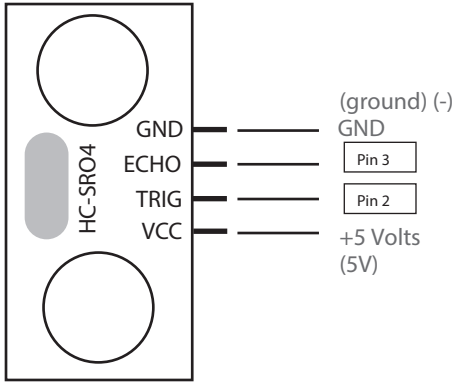
Jumper Wire x8



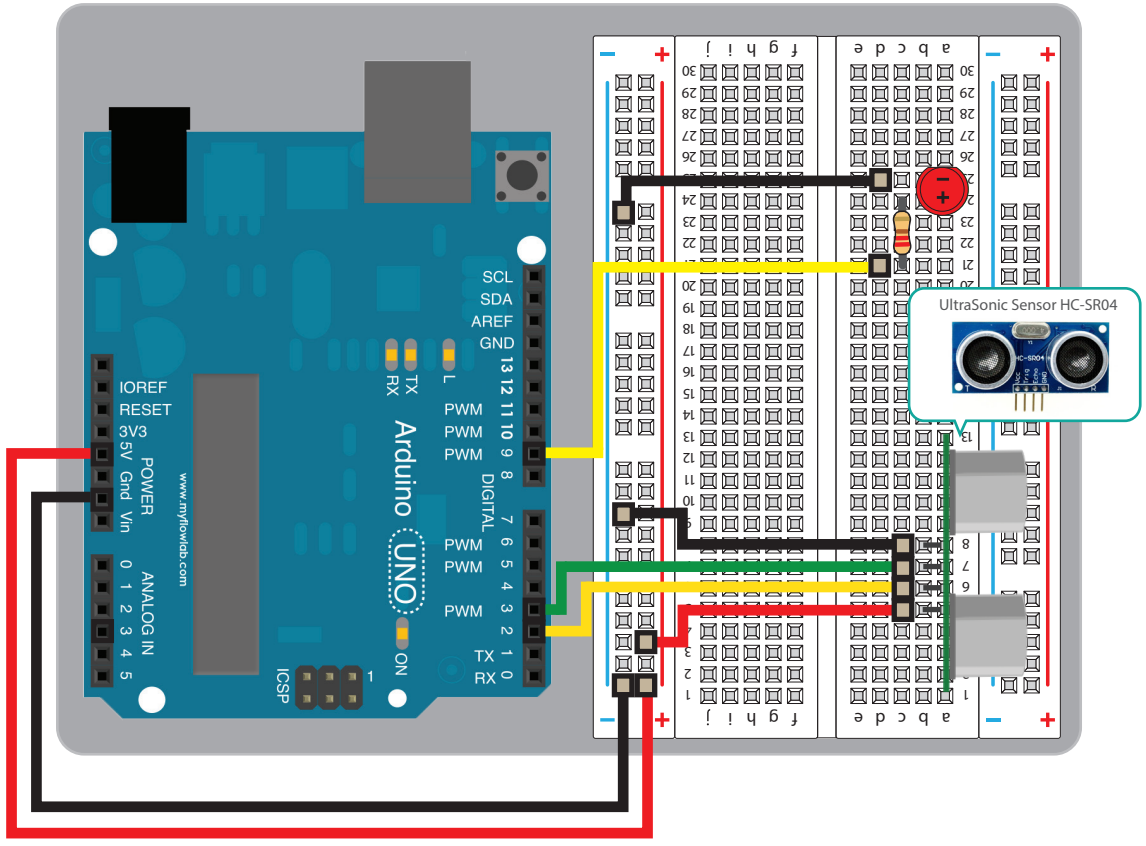
Ultrasonic sensor

SCHEMATIC DIAGRAM

ULTRASONIC SENSOR
HC-SRO4



ULTRASONIC SENSOR HC-SR04 CONNECTION DIAGRAM

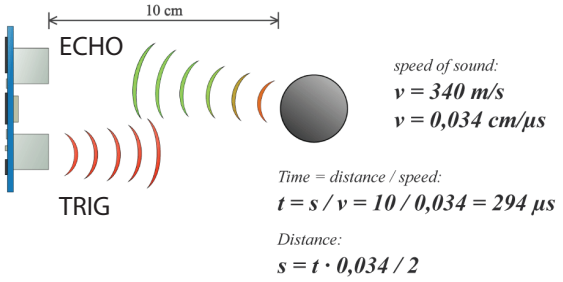


Disconnect Arduino Board USB connection from your PC when carrying out the wires and electronic components connection. Check thoroughly your connections before plugging in the USB cable to your PC.

HOW A ULTRASONIC /SONAR SENSOR WORKS

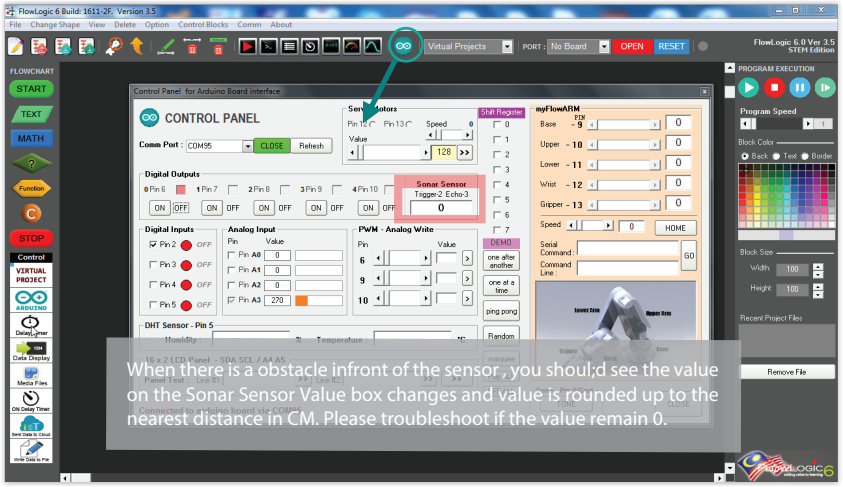
It emits an ultrasound at 40 000 Hz thru TRIG module which travels through the air and if there is an object or obstacle on its path It will bounce back to ECHO module.

For example, if the object is 10 cm away from the sensor, and the speed of the sound is 340 m/s or 0.034 cm/µs the sound wave will need to travel about 294 u seconds. But what you will get from the ECHO modulewill be double that number because the sound wave needs to travel forward and bounce backward. So in order to get the distance in cm we need to devide the received travel time value from the ECHO moduleby 0.034 and divide it by 2.



TESTING YOUR CIRCUIT - ULTRASONIC /SONAR

The Arduino board provides Pin 12 and Pin 13 for Ultrasonic/Sonar sensor TRIG and ECHO pin connection.



TROUBLESHOOTING
3 THINGS TO TRY TO SOLVE THE PROBLEM

1 The value remain 0

Check your connection, please ensure you have connected the VCC to +5volt, GND to GND, TRIG to pin 12 and ECHO to pin 13 on the Ardiuno board

2 Still Not Working

Please ensure the Ultrasonic/Sonar sensor is inserted firmly on the bradboard and facing outward.

3 The value on the Arduino panel not correct

The value is a raw data , you need to program it and apply appropriate formula to achieved the correct distance value (refer to the above)

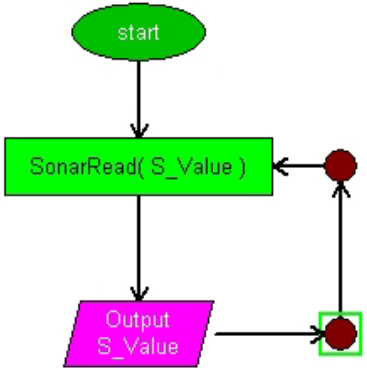
PROGRAMMING EXAMPLES - ULTRASONIC / SONAR SENSOR

The HC-SR04 Ultrasonic/Sonar sensor produces Analog value that has been format to represent distance in centimeter ranging from 2cm to 200cm. Please ensure the sensor is mounted securely and it is 90 degree to base of the breadboard.

You will need to use the SONAR command block from Arduino blocks to read the value from the Ultrasonic/Sonar sensor. Please ensure you have connected the Echo pin of the sensor to Arduino PIN 13 and TRIG pin to Arduino Pin 12.

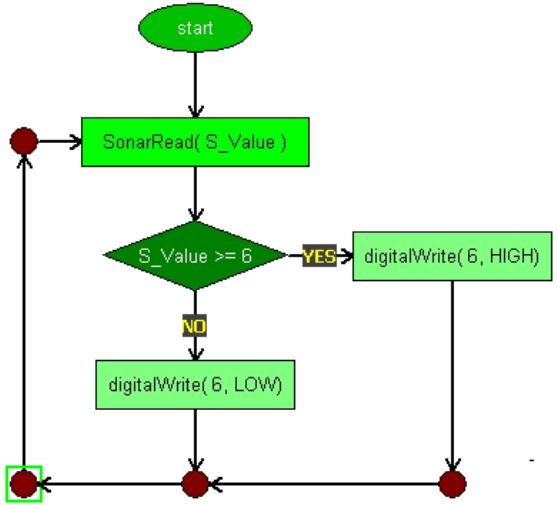
EXAMPLE 1:

Read Sonar Sensor value continuously and display it on the Console window



EXAMPLE 2:

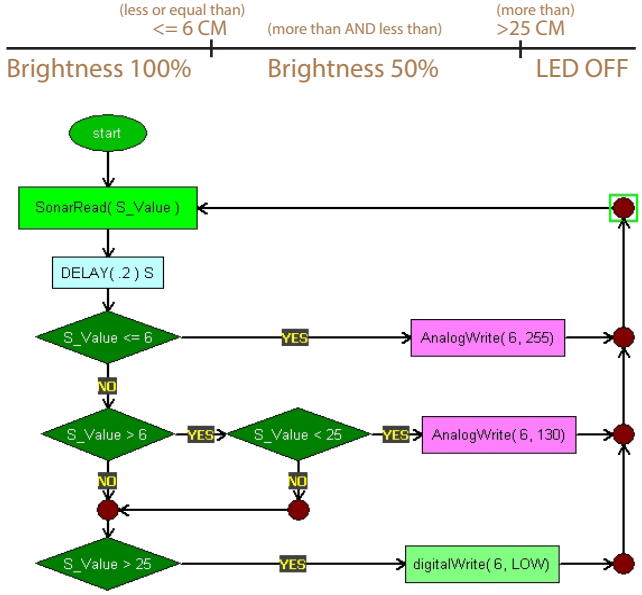
Read Sonar Sensor value and turn on the Output devoce such as LED based on the distance detected.



EXAMPLE 3:

READING MULTIPLE SONAR SENSOR VALUE

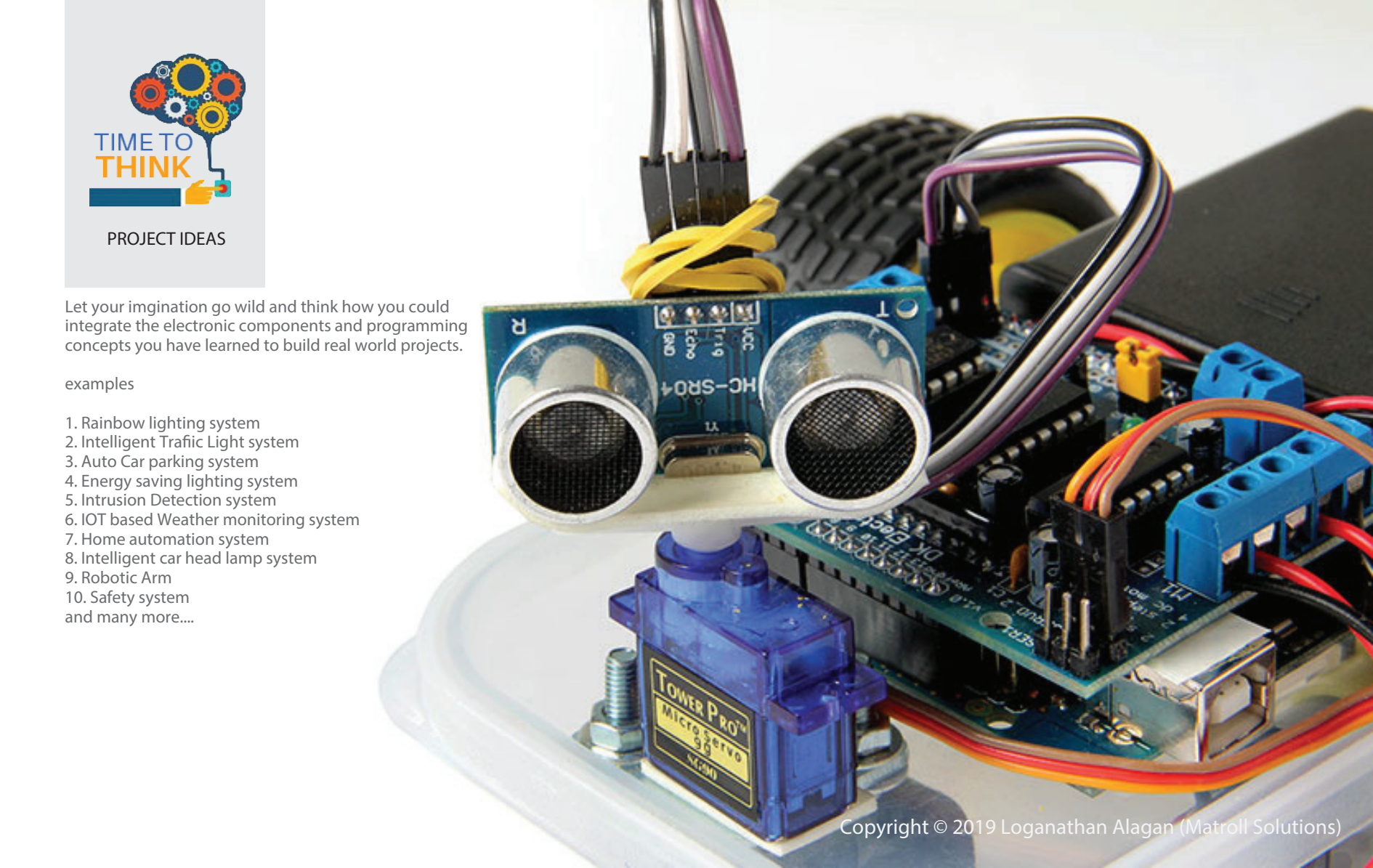
This example will Turn ON and adjust the Brightness of the LED based on distance detected.



Visit us Online:

This is just the beginning of your exploration into Electronics, Programming and Project development. Our website has a wealth of informations and video tutorials to expand your knowledge.

www.myflowlab.com



PROJECT IDEAS

Let your imgination go wild and think how you could integrate the electronic components and programming concepts you have learned to build real world projects.

examples

1. Rainbow lighting system
2. Intelligent Trafiic Light system
3. Auto Car parking system
4. Energy saving lighting system
5. Intrusion Detection system
6. IOT based Weather monitoring system
7. Home automation system
8. Intelligent car head lamp system
9. Robotic Arm
10. Safety system

and many more....

www.myflowlab.com

FlowLogic 6 and myFlowLab Beginner's Guide