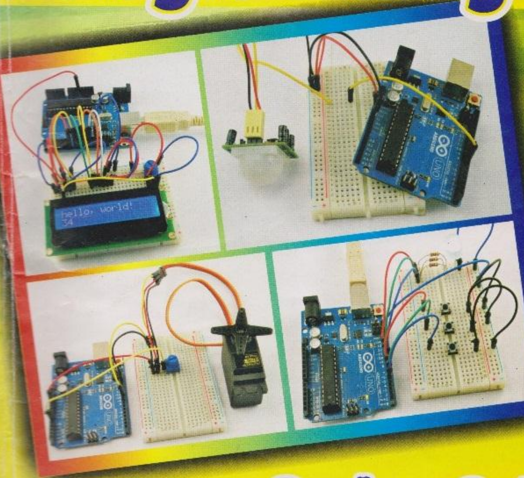
Arduino

Programming



වසන්ත කුමාර

ආර්ඩිනෝ යනු කුමක්ද ?
What is Arduino ?

ආර්ඩිනෝ යනු ඉලෙක්ට්‍රොනික පරිපථ නිර්මාණයේ නියුතු ආධුනිකයන් සඳහා සැකසුනු සරල මෘදුකාංගයකි. මෙම මෘදුකාංගය හා පරිපථ යොදාගනිමින් විවිධ පරිපථ නිර්මානය කලහැකිය. මෙය දැනට බහුලව භාවිතා කරන PIC,IC මෙයික්‍රෝ ක්‍රෝන්ට්‍රෝල හා බොහෝ සමානය.

ආර්ඩිනෝ හා PIC මයික්‍රෝ ක්‍රෝන්ට්‍රෝල සමග සැසඳීමේදී මෙහි ඇති ප්‍රධාන වෙනස්කමක් වන්නේ ආර්ඩිනෝව සඳහා වෙන් වූ පරපථයක් හා මෘදුකාංගයක් ඇති අතර පරිපථ නිර්මාණයේදී මෙම කොටස් ඉතා පහසුවෙන් යොදාගත හැකිවීමයි.

Ardiuno පරිපථය ආධාරයෙන් නිර්මාණ සැකසීමේදී භාවිතා කළහැකි පින් අග ඉවත් වශයෙන් ලබාදී ඇත. එම අගුවලට වයර සවිකිරීමෙන් ඉතා පහසුවෙන් අවශ්‍ය පරිපථය නිර්මාණය කලහැකිය.

එහෙත් PIC,IC මයික්‍රෝ ක්‍රෝන්ට්‍රෝල සඳහා වෙන්වූ පරිපථ පුවරුවක් නොමැති අතර විවිධ මෘදුකාංග මගින් අපට මෙම උපාංගය තුලට දත්ත ලිවිය හැකිය.

Arduino is an open-source electronics prototyping platform based on flexible, easy-to use hardware and software. It is intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments.

ආර්ඩිනෝ පරිපථයන් මගින් කලහැකිදේ කුමක්ද ?

මෙම පරිපථ භාවිතා කර අපට විවිධ නිර්මාණ කලහැකිය. පරිපථයේ පින් අග සඳහා පිටතින් විවිධ සංවේදී උපාංග සවිකල හැකි අතර ඒවාට ලැබෙන දත්ත භාවිතයෙන් නොයෙකුත් කාර්යයන් ස්වයංක්‍රියව සිදුකිරීම සඳහා මෘදුකාංග සැකසිය හැකිය. මේවා අතර,

LED පරිපථ සැකසීම
රොබෝ නිර්මාණ සැකසීම
මෝටර් පාලක පද්ධති සැකසීම
උෂ්ණත්වමාණ සැකසීම
ස්වයංක්‍රීය ආරක්ෂක උපාංග සැකසීම, ප්‍රධාණ තැනක් ගනී.

What Can You Do With an Arduino?

There is a lot you can do with an Arduino. An Arduino can basically do anything by interfacing sensors with a computer. This would allow you to take any sensor and have any action applied with the readings. Such as,

- LED Circuits
- Remote controlled cars
- Motor control circuits
- Temperature sensors
- Home automation systems

ආර්ඩිනෝ පරිපථක් තුල ඇත්තේ කුමක්ද?

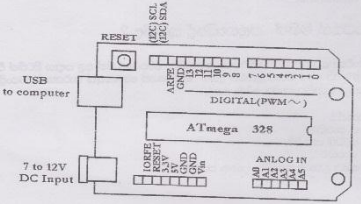
විවිධ වූ ආර්ඩිනෝ පරිපථයන් අපට දැකිය හැකිය. දැනට බහුලව භාවිතා කරණ ආර්ඩිනෝ උනෝ " Arduino UNO" පරිපථය ගැන සොයාබලමු. මෙම පරිපථ සකසන ආකාරය හා ඒවාට මෘදුකාංග ලියන ආකාරය ගැනද සොයා බලමු.

පහත දැක්වෙන්නේ " Arduino UNO" USB පරිපථ සටහනකි.

ඔබට Atmega 168/328p මයික්‍රෝ කොන්ට්‍රෝලය ඇතුලත් " Arduino UNO" පරිථයක් වෙළඳ පොළෙන් මිලදී ගතයුතුය. මෙය ආධාරයෙන් සරල ප්‍රෝග්‍රෑම් සකසාගන්නා ආකාරය සොයා බලමු.

වෙළඳ පොළෙන් මිලදීගැනීමට ඇති " Arduino UNO" USB කොන්ට්‍රෝල පරිපථයේ පහත දැක්වෙන පරිපථ උපාංග දැකිය හැකිය

- Processor: 16 Mhz ATmega328 / Atmega186
- Flash memory: 32 KB
- Ram: 2kb
- Number of analog inputs: 6
- Number of digital I/O: 14 (6 pwm)



ලබාදී ඇති CD තැටියේ Arduino UNO" USB පරිපථ සටහ ණ් Arduino UNO" PCB ලෙස සඳහන් Folder යේ ලබාදී ඇත.

" Arduino UNO" PCB පුවරුවේ ඇනලොග් (Analog Input) ලෙස අග්‍ර 6 ක් තිබේ. ඒවා A0,A1,A2,A3,A4,A5 ලෙස දක්වා ඇත. මෙම අග්‍ර ඇනලොග් සංවේදි උපාංග (Analog Sensors) සවිකිරීම සඳහා යොදාගත හැකිය.

උදාහරණයක් ලෙස උෂ්ණත්වමාන (DS1820 LM35) , LDR , IR Sensors , වැනි උපාංග

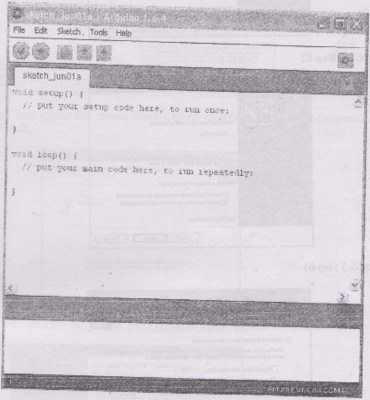
Digital Input ලෙස 0 සිට 13 දක්වා අග්‍ර ලබාදී අත. මෙම අග්‍ර වලින් විවධ වූ උපාංග ක්‍රියා කරවීමට අවශ්‍ය විභවයන් ලබාගැනීමට හා ලබාදීමට හැකිය. මෙහි PWM (Pulse With Modulation) ලෙස ලබාදී ඇති අග්‍ර ඉදිරියෙන් මෙම සංකේතය " ~" සටහන් කර ඇත. මේ අනුව මෙහි PWM අග්‍ර 6කි.

පරිපථයේ 13 වන පින් අග්‍රයට PCB පුවරුවේ LED බල්බයක් ඇඳුම් කර තිබේ.

පරිපථයට දත්ත ලබාදීමේදී USB වයරය යොදාගතහැකි අතර මෙම වයරය භාවිතයෙන් වුවත් පරිපථය ක්‍රියා කරවිය හැකිය. නැතහොත් පරිපථය ප්‍රෝග්‍රෑම් කිරීමෙන් පසුව USB වයරය ඉවත් කර 7-12V,DC බල සැපයුමක් මගින් වුවද පරිපථය ක්‍රියා කරවිය හැකිය.

ආර්ඩිනෝ මෘදුකාංගය

මෙම මෘදුකාංගය ලබාදී ඇති CD තැටියේ ARDUINO IDE ලෙස සඳහන් ලිපිගොනුවේ ඇතුලත්වේ. එය ඔබගේ පරිගනකයට ඇතුලත් කරගන්න. නිවැරදිව මෘදුකාංගය ඇතුලත් වුවහොත් ඔබට Desktop හි Ardiuno ලෙස අයිකනයක් දැකිය හැකිය. මෙය මවුසය මගින Open කල විට පහත දැක්වෙන Arduino 1.6.4 හෝ Arduino 1.6.5 මෘදුකාංගය ලැබෙයි.



ඔබ මිළදීගත් " Arduino UNO" PCB පුවරු පරිගනකයට USB වයරයෙන් සම්බන්ධ කරන්න.

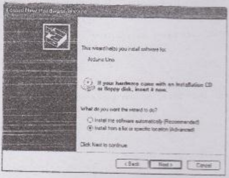
දැන් පහත දැක්වෙන ආකාරයට " Arduino UNO" PCB පුවරු සඳහා පරිගනකයට Drivers ලෝඩ් කරගත යුතුය.

මේ සඳහා ප්‍රථමයෙන් Divice Manager වෙතගොස් add new devices මගින් Arduino UNO" USB පරිපථය පරිගනකය හා සම්බන්ධ කිරීමට පහත දැක්වෙන පියවර අනුගමනය කරන්න.

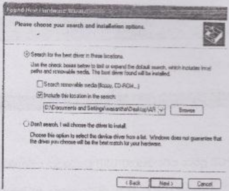
දැන් පහත දැක්වෙන ආකාරයට මෙනු සකස්කරගන්න.

පියවර 1 Step 01

පියවර 2 Step 02



පියවර 3 Step 03



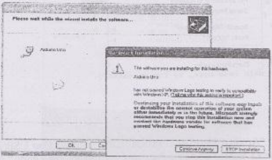
ඉහත සඳහන් මෙනුව සැකසීමෙන් පසුව පහත දැක්වෙන ආකාරයට මෙම මෙදුකාංගය ක්‍රියාකිරීම සඳහා අවශ්‍ය Drivers තෝරා ගතයුතුය.

1. Browse වෙත ගොස් ඔබ ARDIUNO මෘදුකාංග පරිගණකයේ තැම්පත් කල Folder ය Open කරන්න.

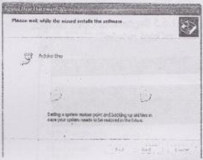
2. මෙම Folder යේ ඇති Driver ලෙස සදහන් Folder ය open කරන්න.

3. Driver Folder යේ ඇති FIDI USB Driver ලෙස සදහන් Folder ය Select කර next බටනය ඔබන්න.

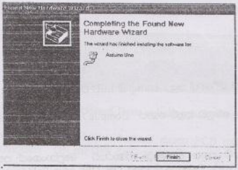
4. දැන් පහත දැක්වෙන මෙනුව ලැබෙයි. එහි ඇති Continue Anyway බටනය ඔබන්න



5. Ardiuno Uno සඳහා වන drivers පහත රූපයේ දැක්වෙන ආකාරයට පරිගණකයට ඇතුල්වීම ආරම්භවෙයි.



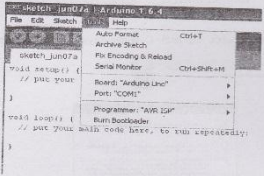
එය නිවැරදිව සිදුවුයේනම් පහත දැක්වෙන මෙනුව ලැබෙයි



දැන් මෙම මෙනුවේ Finish බටනය ඔබා මෙම ක්‍රියාවලියෙන් ඉවත්වන්න.

ඉහත දැක්වෙන ආකාරයට Ardiuno Uno, PCB පුවරුව පරිගණකයට නිවැරදිව ඇතුලත් වී ඇතිදැයි බලමු.

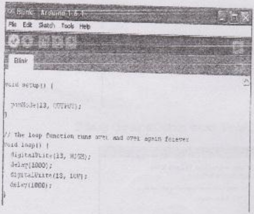
Arduino 1.6.4 මෘදුකාංගය Open කරන්න. Tools මෙනුව මවුසය මගින් Open කල විට ලැබෙන මෙනුව පහත රූපයේ දැක්වෙන ආකාරයට සැකසී ඇති දැයි බලන්න.



Tools මෙනුව යටතේ Board ; "Arduino Uno" ලෙස දැක්වේ නම් Arduino Uno USB ,PCB පරිපථය පුවරුව අදාල Drivers පරිගණකය මගින් නිවැරදිව ලබාගෙන ඇති බවත් එය Port; " COM1" වෙත ඈදුම් වී ඇති බවත් පෙනේ. (ඔබගේ පරිගණකයේ දී මෙම Com1 අගය වෙනස්විය හැකි බව සලකන්න)

ප්‍රෝග්‍රෑම (Sketch) Arduino Uno PCB පුවරුව තුලට ඇතුල් කිරිම

1. පළමුව Arduino 1.6.4 හෝ Ardiuno 1.6.5 මෘදුකාංගය Open කරන්න.
2. දැන් පහත දැක්වෙන ප්‍රෝග්‍රෑමය එහි සටහන් කරන්න.



ඉහත ප්‍රෝග්‍රෑමය නිවැරදිව සටහන් කිරීමෙන් පසු Desktop හි LED ON OFF ලෙස save කරන්න.

දැන් මෘදුකාංගයේ ඇති Sketch මෙනුව තුලට ගොස් Compile / veryfy හෝ Ctrl +R බටනය ඔබන්න.

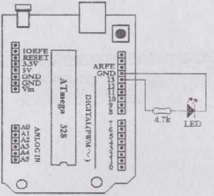
ලියා ඇති Sketch ය නිවැරදිනම් පහත දැක්වෙන ආකාරයට මෘදුකාංගයේ පහල කොටසේ Done compilling ලෙස දැක්වෙන අතර Desktop හි LED_ON_OFF ලෙස Folder තුල ඉහත අප ලියු Sketch ය දැකිය හැක.

දැන් USB PCB පුවරුව (Arduino Uno පරිපථය) පරිගණකය හා සම්බන්ධ කරන්න. මෙම මෘදුකාංගයේ ඇති Upload ලෙස සඳහන් බටනය හෝ Ctrl +U බටනය එබීමෙන් මෙම sketch ය Arduino Uno පරිපථයේ ඇති IC උපාංගය වෙත යැවීම ආරම්භවෙයි.

නිවැරදිව Upload වුයේ නම් පමණක් Done uploding ලෙස මෘදුකාංගයේ පහළින් දැක්වෙයි.

දැන් පහත දැක්වෙන ආකාරයට පරිපථයක් සකසා PCB පුවරුවට DC විභවයක් ලබාදුනහොත් බල්බය නොනවත්වා ON හා OFF වීමට පටන්ගනී.

මේ ආකාරයට සකසූ Sketch හෝ ප්‍රෝග්‍රෑම ප්‍රථමයෙන් Compile කිරීමත් එය නිවැරදිනම් දෙවනුව Arduino Uno පරිථයට upload කිරීමත් කළ හැකිය.

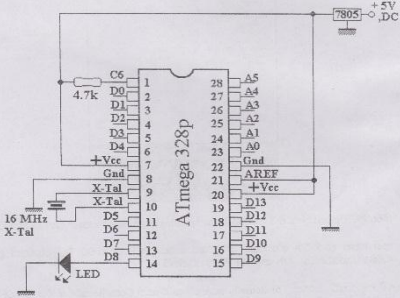


Ardiuno Uno පරිපථයට Upload කර ගත් Sketch ය තැම්පත්වන්නේ Atmega 328p, IC උපාංගය තුලය.

පරිපථ සැකසීමේදී මෙම Ardiuno Uno PCB පරිපථ කොටස යොදාගතහැකි වුවත් මෙම කොටස මිල අධික වන බැවින් Ardiuno Uno පරිපථයේ ඇති Atmega 328p, IC උපාංගය PCB පුවරුවෙන් ගලවා සැකසීමට අවශ්‍ය පරිපථයට යොදාගතහැකිය.

Ardiuno Uno PCB යට නව Atmega 328p IC උපාංගයක් ඇතුලත් කිරීමේදී එය එක් වරක් Boothload කිරීමේ ක්‍රියාවලියකට භාජනය කිරීමෙන් පසු Ardiuno Uno PCB පුවරුවට සවිකල හැකිය.

උදාහරණයක් ලෙස ඉහත sketch මගින් ක්‍රියාකරන LED බල්බය සඳහා පරිපථයක් සැකසීමේදී Ardiuno Uno පරිපථයේ ඇති Atmega 328p ගලවා පහත දැක්වෙන ආකරයට පරිපථය සැකසිය හැකිය.

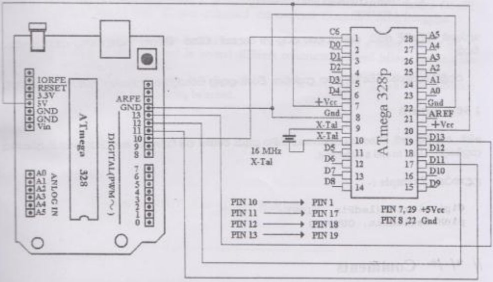


BOOTHLOADING කිරීම යනු කුමක්ද ?

- Ardiuno Uno පරිපථයේ ඇති ATmega328 / Atmega168 ,IC උපාංගතුලට සකසන ප්‍රෝග්‍රෑමය හෝ sketch ය ඇතුලත් කලහැකි ආකාරයට IC උපාංගය සැකසීම Boothloading කිරීම ලෙස හඳුන්වයි.
-
- IC උපාංගය Boothload කිරීමේදී ATmega ,IC උපාංගය තුල දත්ත ගබඩා කිරීමට වෙන්වූ Flash Memory කොටසේ Ardiuno මෘදුකාංගයෙන් ලියනු ලබන ප්‍රෝග්‍රෑම තැම්පත් කිරීම සඳහා Memory කොටසක් වෙන් වෙයි.
-
- වෙළඳපොලෙන් මිලදීගන්නා Ardiuno Uno පරිපථවල ඇති IC උපාංග Boothload කර ඇති බැවින් එය නැවතත් boothload කලයුතු නැත.
-
- එහෙත් පරිපථයේ ඇති Atmega 328p / Atmega168 ,IC උපාංගය ඉවත්කර මිලදීගත් නව IC උපාංගයක් පරිපථයට ඇදුම් කරන අවස්ථාවලදී එය ප්‍රථමයෙන් එක් වරක් පහත දැක්වෙන ආකාරයට Boothload කලයුතුය.

Atmega 168/328p, IC උපාංගයක් Boothload කරගන්නා ආකාරය ලබාදී ඇති CD තැටියේ Boothload IC ලිපිගොනුව තුල දක්වා ඇත.

පහත දැක්වෙන්නේ Atmega 168/328p, IC උපාංගයක් Boothload කිරීම සඳහා යොදාන්නා පරිපථයකි.



Arduino මෘදුකාංගයට ප්‍රෝග්‍රෑම ලිවීම

Arduino මෘදුකාංගයේ ලියන ප්‍රෝග්‍රෑම sketch යක් ලෙස හඳුන්වයි. මෙම sketch ලිවීමේදී එවා පහලට ගලායන ආකාරයට ලිවිය යුතුය. මේ ආකාරයට ලියන sketch හෝ ප්‍රෝග්‍රෑමය කොටස් 4 කට බෙදිය හැකිය.

1. ප්‍රෝග්‍රෑමය ලිවීම ආරම්භයේදී Header ලෙස කොටසක් ලියයි. මෙම කොටසින් ප්‍රෝග්‍රෑමය සකසන අයගේ නම, සැකසූ දිනය හෝ එය සකසන්නේ කුමක් සඳහාද යන්න පසුව දැනගැනීමට අවශ්‍යනම් පමණක් එය ලියාතැබිය හැකිය. මෙම කොටස ප්‍රොග්‍රෑමයට ඇතුලත් නොවේ.

2. Arduino පරිපථයට ඈඳුම් කරන උපාංග සවිවන පින් අග්‍ර හඳුන්වා දීම.

3. Voild setup( ) යටතේ මෙම පින් අග්‍ර ක්‍රියාකරන ආකාරය.
   මෙම ප්‍රෝග්‍රෑම් කොටස පරිපථය ක්‍රියාත්මක වන විට එක් වරක් පමනක් මෛයික්‍රෝ කොන්ට්‍රෝලය මගින් කියවයි.

4. Voild loop( ) යටතේ මෙම පරිපථය ක්‍රියාත්මක වියයුතු ආකාරය විවිධ විධාන භාවිතයෙන් දැනුම්දෙයි.

sketch හෝ ප්‍රෝග්‍රෑම ලිවීමේදී ලබාදෙන විධානයන් ප්‍රධාන වශයෙන් කොටස් 3 කට බෙදිය හැකිය. මෙම කොටස් එක් කර අවශ්‍ය කාර්යය සිදුකිරීම සඳහා ප්‍රෝග්‍රෑමයක් හෙවත් sketch යක් සැකසිය හැකිය.

මෙම කොටස් 3 න පහත පරිදිවේ

1.Structure
2.*Values* (variables and constants),
3. *Functions*.

ප්‍රෝග්‍රෑම ලිවීමේදී අපට හමුවන ඉතා සරල හා වැදගත් විධාන කිහියක් යොදාගන්නා ආකාරය සොයා බලමු .

පළමුව ප්‍රෝග්‍රෑම ලිවීමේදී පහත දැක්වෙන විරාම ලකුණු නිවැරදිව යොදාගන්නා ආකාරය සොයාබලමු.

; semicolon

මෙම නැවතිම සෑම විධානයක් අවසානයේ තිබිය යුතුය. එමගින් එම විධානයේ හෝ Statement හි අවසානය දැනුම්දෙයි. Used to end a statement.

උදාහරණ Example :-

```
digitalWrite(ledPin, HIGH);
pinMode(ledPin, OUTPUT);
```

// */ /* Comments

ප්‍රෝග්‍රෑමයක් ලිවීමේදී යම් පේලියක හෝ ප්‍රෝග්‍රෑම් කොටසක් මගින් වන කාර්යය කුමන්දැයි කියා දැනගැනීමට පහත දැක්වෙන ආකාරයට එවා ලියාතැබිය හැකිය. මෙම විස්තර ප්‍රෝග්‍රෑමයෙන් වෙන් කරතැබීම පහත දැක්වෙන ආකාරයට කලහැකිය.

Comments are lines in the programme that are used to inform yourself or others about the way the program works. They are ignored by the compiler, and not exported to the processor, so they don't take up any space on the Atmega chip.
Comments only purpose are to help you understand (or remember) how your program works or to inform others how your programme works. There are two different ways of marking a line as a comment.

උදාහරණ Example :-

```
x = 5; // This is a single line comment. Anything after the slashes is a comment
       // to the end of the line
```

දීර්ඝ පෙළ වාක්‍යයක් ලිවීමේදී එය පහත දැක්වෙන ආකාරයට ලියන්න /*

```
/* this is multiline comment - use it to comment out whole blocks of code

if (gwb == 0){   // single line comment is OK inside a multiline comment
x = 3;           /* but not another multiline comment - this is invalid */
}
// don't forget the "closing" comment - they have to be balanced!
*/
```

{ } Curly Braces

{} සහල වරහන් භාවිතය මෙම මාදුකාංගයේදී ඉතා වැදගත්වේ. මෙහිදී යම් විධානයක් ආරම්භ කිරීමේදී මෙම වරහන ඇතුලත් කලයුතු අතර විධානය අවසානයේ එය නැවතත් වරහනකින් වසාදැමිය යුතුය.

Curly braces (also referred to as just "braces" or as "curly brackets") are a major part of the C programming language. They are used in several different constructs, outlined below, and this can sometimes be confusing for beginners.
An opening curly brace "{" must always be followed by a closing curly brace "}". This is a condition that is often referred to as the braces being balanced.

උදාහරණ Example :-

```
void setup()
{
pinMode(buttonPin, INPUT); // button pin 3 is an input
}
.........................................................................
void setup()
{
pinMode(buttonPin, INPUT);
pinMode(led, outPUT);
}
```

include

ප්‍රෝග්‍රෑම ලිවීම පහසු කිරීම සඳහා මෙම විධානය යොදාගනී. මෙම මෘදුකාංගයේ විශේෂිත උපාංග ක්‍රියාකරවීම සඳහා සැකසු ප්‍රෝග්‍රෑම කොටස් Ardiuno මෘදුකාංගයේ ඇති library යක තැම්පත්කර ඇත. මෙම library කොටස් ප්‍රෝග්‍රෑමයට එක් කිරීම මෙම විධානයෙන් සිදුකළහැකිය.

#include is used to include outside libraries in your sketch. This gives the programmer access to a large group of standard C libraries (groups of pre-made functions), and also libraries written especially for Arduino.
The main reference page for AVR C libraries (AVR is a reference to the Atmel chips on which the Arduino is based)
Note that **#include**, similar to **#define**, has no semicolon terminator, and the compiler will yield cryptic error messages if you add one.

උදාහරණ Example :-

`#include <Stepper.h>` // මෙම විධානයෙන් Stepper මෝටර ක්‍රියාකරවීම සඳහා සැකසුනු ප්‍රෝග්‍රෑම කොටස Library යෙන් ලබාගනී.

`#include <Servo.h>` // මෙම විධානයෙන් servo මෝටර ක්‍රියාකරවීම සඳහා සැකසුනු ප්‍රෝග්‍රෑම කොටස Library යෙන් ලබාගනී.

`#include <LiquidCrystal.h>` // මෙම විධානයෙන් LCD Display ක්‍රියාකරවීම සඳහා සැකසුනු ප්‍රෝග්‍රෑම කොටස Library යෙන් ලබාගනී.

pinMode()

pinMode() මගින් Arduino පරිපථයට ඇදුම්කරන උපාංග හැසිරෙනුයේ පරිපථයට Input දත්ත ලබාගන්නා ආකාරයටද නැතහොත් Output ලෙස දත්ත ලබාදීමටද යන්න මෙම විධානය මගින් සිදුකරයි.

Configures the specified pin to behave either as an input or an output. See the description of <u>digital pins</u> for details on the functionality of the pins.

උදාහරණ Example :-

pinMode(pin, mode)

```
pinMode(ledPin, OUTPUT); // sets the digital pin as output
pinMode(button, INPUT);  // sets the digital pin as output
pinMode(buzzer, OUTUT);  // sets the digital pin as output
```

digitalWrite()

digitalWite() මගින් Arduino පරිපථයට ඇදුම්කරන උපාංග සවිවන පින් අග්‍ර හැසිරිය යුතු අකාරය හඳුන්වා දෙයි. මෙහිදී digitalWite මගින් යම් පින් අග්‍රයක 0 හා 5V වෝල්ටියතා අතර වෙනස්වන බව දැනුම්දෙයි.

If the pin has been configured as an OUTPUT with pinMode(), its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for HIGH, 0V (ground) for LOW.
If the pin is configured as an INPUT, digitalWrite() will enable (HIGH) or disable (LOW) the internal pullup on the input pin. It is recommended to set the pinMode() to INPUT_PULLUP to enable the internal pull-up resistor.

උදාහරණ Example :-

digitalWrite(pin, value)

```
digitalWrite(ledPin, HIGH);   // sets the LED on 5V
digitalWrite(ledPin, LOW);    // sets the LED off 0v
```

digitalRead()

digitalRead() මගින් Arduino පරිපථයට ඇදුම්කරන සංවේදී උපාංග (sensor) සවිවන පින් අග්‍රයේ වෝල්ටීයතාවය (0V හෝ 5V) කියවා බලයි.
Reads the value from a specified digital pin, either HIGH or LOW.

උදාහරණ Example :-

digitalRead(pin)

```
val = digitalRead(inPin); // read the input pin
digitalWrite(ledPin, val); // sets the LED to the button's value
```

setup( )

Setup Function මගින් **Arduino** පරිපථයට ඇදුම්කරන උපාංග හා ඒවා සම්බන්ධවන පින් අග්‍ර පිළිබඳව මෘදුකාංගයට දැනුම්දීම මෙහි මූලික කාර්යය වේ. මෙම කොටස පරිපථයේ ක්‍රියා කාරිත්වය ආරම්භ වන අවස්ථාවේදී එක් වරක් පමණක් මෛක්‍රෝ කොන්ට්‍රෝලය මගින් කියවනු ලබයි.

Function is called when a sketch starts. Use it to initialize variables, pin modes, start using libraries, etc. The setup function will only run once, after each power up or reset of the Arduino board.

උදාහරණ Example :-

```
int buttonPin = 3; //button connect to pin 3

void setup()
{
pinMode(buttonPin, INPUT); // බටනය මගින් INPUT ස්වින්ද ලබාදෙන බව දන්වයි
                              button pin 3 is an input
}

int led = 13; // led connect to pin 13

void setup() {
pinMode(LED, OUTPUT); // LED බල්බය Output ලෙස ක්‍රියාකරන බව දන්වයි
```

```
}
int buttonPin  = 3;
int led        = 13;
void setup()
{
pinMode(buttonPin, INPUT);
pinMode(led, outPUT);
}
```

මෙම කොටස තුල පරිපථය ඇඳුම්කර ඇති උපාංග කුමන ආකාරයෙන් ක්‍රියාකරනු ලබන්නේදැයි සටහන් කලයුතුය.

සරලව කිව හොත් යම් බටනයක් (Input ස්වන්දයක් ලබා දෙන) හෝ LED බල්බයක් (Output වෝල්ටියතාවයක් පරිපථයෙන් ලබාදෙන) වැනි උපාංගයක් පරිපථයට සවිකර ඇති පින් අග්‍රය මාදුකාංගයට දැනුම්දීමට යොදා ගනී.
මෙම Int ලෙස හදුන්වන්නේ Integer Value අගයක් හෙවත් පූර්ණ සංඛ්‍යා අගයක් බවයි.

loop()

මෙම Loop Function විධාන කොටස මගින් යම් කාර්යයක් නොනවත්වා චක්‍රයක් (Loop) ලෙස සිදුකරයි.

After creating a setup() function, which initializes and sets the initial values, the loop() function does precisely what its name suggests, and loops consecutively.

උදාහරණ Example :-

```
int led = 13; // setup initializes serial and the button pin
void setup()
{
  pinMode(led, output); // }
>>>>>>>>>>>AA<<<<<<<<<<<<<
void loop()
{
digitalWrite(13, HIGH);//මෙම විධානයෙන් පින් අග්‍ර13 සම්බන්ධ උපාංගයට  5V විභවයක් ලබාදෙන ලෙස දන්වයි
delay(1000);// ඩිලේ කාලය බල්බය ON වේ
digitalWrite(13, LOW);// මෙම විධානයෙන් පින් අග්‍ර13 සම්බන්ධ උපාංගයට 0V විභවයක් ලබාදෙන ලෙස දන්වයි
delay(1000); // ඩිලේ කාලය බල්බය OFF වේ
}     >>>>>>>>BB<<<<<<<<<<
```

Loop () විධානයට අයත් AA සිට BB දක්වා වූ කාර්යය වන Led බල්බය ON හා OFF වීම නොනවත්වා සිදුකරයි.

if (conditional) and ==, !=, <, > (comparison operators)

if විධානය ආධාරයෙන් පරිපථයේ යම් පින් අග්‍රයනට හෝ කිහිපයකට දත්තයක් ලැබී ඇත්නම් එවිට පමනක් වෙනත් කාර්යයක් සිදුකලයුතු බව මාදුකාංගයට දැනුම්දීමට යොදාගතහැකිය.

මෙහිදී පරිපථය සම්බන්ධ බටනයක් එබී ඇතිනම් පමනක් පරිපථයට සවිකර ඇති LED බල්බයක් ON කිරීම හෝ ON වී ඇති යම් උපාංගයක් OFF කිරීම සඳහා මෙම if විධානය යොදාගතහැකිය.

if, which is used in conjunction with a comparison operator, tests whether a certain condition has been reached, such as an input being press,
The program tests to see if some Variable is greater than 50. If it is, the programme takes a particular action. Put another way, if the statement in parentheses is true, the statements inside the brackets are run. If not, the programme skips over the code.
The brackets may be omitted after an *if* statement. If this is done, the next line (defined by the semicolon) becomes the only conditional statement.

උදාහරණ Example :-

```
int led1 = 13; // setup initializes serial and the led pin
int led2 = 12;
int button = 1
void setup(){
  pinMode(led1, output); //
  pinMode(led2, output); //
  pinMode(button, input); //
}
 >>>>>>>>>>>>AA<<<<<<<<<<<<<
if (button = 1){
  digitalWrite(LED1, HIGH);
  digitalWrite(LED2, HIGH);
}
>>>>>>>>>>>>BB<<<<<<<<<<<<<
```

ඉහත ප්‍රෝග්‍රෑමය ලිවීමේදී Led1 උපාංගය ,led2 උපාංගය හා button තය පරිපථයේ සවිවන පින් අග්‍ර මෘදුකාංගයට පළමුව දැනුම් දී ඇත.
දෙවනුව ඒවා යේ ක්‍රියාකාරිත්වය void setup() යටතේ දන්වයි.
If විධානය යටතේ Button ය එබී ඇතිනම් (1 හෝ High) පරිපථයට සම්බන්ධ Led1,Led2 ON විය යුතුබව දන්වයි.

If විධානය පහත දැක්වෙන ආකාරයටද යොදාගත හැකිය.

මෙහිදී විචල්‍යයක් යම් පූර්ණ සංඛ්‍යා වකට සමාන , විශාල හෝ කුඩා වන විට පමණක් යම් කාර්යයක් සිදු කලයුතු බව දැනුම්දීමට යොදාගත හැකිය.

```
if (x = 120) digitalWrite(LEDpin1, HIGH);
if (x > 120) digitalWrite(LEDpin2, HIGH);
if (x < 120){ digitalWrite(LEDpin3, HIGH);
if (x = 120)
{
  digitalWrite(LEDpin1, HIGH);
  digitalWrite(LEDpin2, HIGH);
}
```

if / else

If විධානය මගින් පළමුව පරිපථයේ යම් පින් අග්‍රයනට හෝ කිහිපයකට input සංඥාවක් ලැබී ඇතිදැයි පරීක්ෂා කරයි. එසේ ලැබී ඇතිනම් පමණක් වෙනත් කාර්යයක් සිදුකලයුතු බව මෘදුකාංගයට දැනුම් දෙයි.

else විධානය මගින් if විධානය යටතේ කලයුතු කාර්යය සිදුවී නැතිනම් else if විධානය යටතේ වූ වෙනත් කාර්යයක් සිදු කලයුතුබව මෘදුකාංගයට දැනුම්දෙයි. මේ ආකාරයට If හා else if විධානයන් භාවිතා කළහැකිය.

if/else allows greater control over the flow of code than the basic **if** statement, by allowing multiple tests to be grouped together.

else can proceed another **if** test, so that multiple, mutually exclusive tests can be run at the same time. Each test will proceed to the next one until a true test is encountered.

When a true test is found, its associated block of code is run, and the programme then skips to the line following the entire if/else construction. If no test proves to be true, the default **else** block is executed, if one is present, and sets the default behavior.

Note that an **else if** block may be used with or without a terminating **else** block and vice versa. An unlimited number of such **else if** branches are allowed.

උදාහරණ Example :-

```
if (x < 500)
{
  digitalWrite(LEDpin1, HIGH);
}
else if (x= 1000)
{
  digitalWrite(LEDpin2, HIGH);
}
else if (x= 1500)
{
  digitalWrite(LEDpin3, HIGH);
}
else
{
digitalWrite(LEDpin4, HIGH);
}
```

for statements

for විධානය වරහන් තුළ ලියා ඇති යම් විධානයක් නැවත නැවත ගණනය කරීම සඳහා යොදාගතහැකිය.

The **for** statement is used to repeat a block of statements enclosed in curly braces. An increment counter is usually used to increment and terminate the loop. The **for** statement is useful for any repetitive operation, and is often used in combination with arrays to operate on collections of data/pins.

There are three parts to the **for** loop header:

```
for (initialization; condition; increment) {
```

උදාහරණ Example :-

බල්බයක දීප්තිය ක්‍රමයෙන් ඉහලට හා පහලට ගෙනයන ප්‍රෝග්‍රෑමයක් සකසමු.

```
// Dim an LED using a PWM pin
int PWMpin = 10; //
void setup()
{  // no setup needed
  }
```

```
void loop()

{ (initialization; condition; increment)
  for   (int i=0;           i <= 255;          i++){

      analogWrite(PWMpin, i);
        delay(10); } }
```

මෙහිදී int i යන පූර්ණ සංඛ්‍යාව 0 වට සමාන කරයි. දැන් එය 1 කින් ඉහල යවා එම අගය 255 යන අගයට වඩා විශාල හෝ සමාන දැයි බලයි. ඒසේ සමාන නොවුනහොත් එම අගය PWMpin අග්‍රය ඉවත යොමු කර Led බල්බයේ දීප්තිය ක්‍රමයෙන් ඉහලට ගෙනයයි. මෙහිදී සිදුවන්නේ LED බල්බය සම්බන්ධ පින් අග්‍රයේ වෝල්ටීයතාවය 0 සට 5V දක්වා ක්‍රමයෙන් වැඩිමයි.

පහත දැක්වෙන ප්‍රෝග්‍රෑමයෙන් LED බල්බයක දීප්තිය ක්‍රමයෙන් පහලට ගෙනයයි.

```
void loop()
{
   int x = 1;

   for (int i = 0; i > -1; i = i + x){

      analogWrite(PWMpin, i);

      if (i == 255) x = -1;

      delay(10);
   }
}
```

while loops

මෙම while loop විධාන කොටස මගින් යම් කාර්යයක් නොනවත්වා චක්‍රයක් ලෙස දිගටම සිදුකරගෙන යන අතර එයට ලබාදී ඇති condition යක් හා සමාන, විශාල හෝ කුඩා වූ විට එම චක්‍රයෙන් ඉවත්වී අදාල ඊළඟ කාර්යය සිදුකරයි.

while loops will loop continuously, and infinitely, until the expression inside the parenthesis, () becomes false. Something must change the tested variable, or the **while** loop will never exit. This could be in your code, such as an incremented variable, or an external condition, such as testing a sensor.

```
while(expression){

   // statement(s)
}
```

උදාහරණ Example :-

```
var = 0;
while(var < 200){
 digitalWrite(LEDpin1, HIGH); // do something repetitive 200 times
  var++;
}
var = 0;
while(var < 100){

digitalWrite(LEDpin1, HIGH); // do something repetitive 100 times
digitalWrite(LEDpin2, HIGH); // do something repetitive 100 times
  var++;
}
```

do – while

මෙම do විධාන කොටස මගින් යම් කාර්යයක් ආරම්භකර එම කාර්යයට while විධානය යටතේ දක්වා ඇති වාර ගණනක් සිදුකර එය නවතා දමයි. මේ අනුව මේ ආකාරයට ලියනු ලබන ප්‍රෝග්‍රෑමයක් එක්වරක් පමණක් සිදුවෙයි. එය නැවතත් සිදුකිරීමට පරිපථය නැවත ක්‍රියාත්මක කළයුතුය.

The **do** loop works in the same manner as the **while** loop, with the exception that the condition is tested at the end of the loop, so the **do** loop will *always* run at least once.

```
do
{
    // statement block
} while (test condition);
```

උදාහරණ Example :-

```
do
{
  delay(50);           // wait for sensors to stabilize
  x = readSensors();   // check the sensors

} while (x < 100); මෙය 100 වතාවක් සිදිකර ප්‍රෝග්‍රෑමය අවසාන කරයි
```

switch / case statements

මෙම switch ...case විධාන කොටස මගින් ප්‍රෝග්‍රෑමයක පහතට ගලායෑම පාලනය කරන අතර ප්‍රෝග්‍රෑමයේ විවිධ ස්ථානවලදී ලැබෙන විධාන ක්‍රියාත්මක කිරීම මෙමගින් සිදුවෙයි. switch statement මගින් csed Statement හි ඇති විචල්‍ය අගයක ඇති අගය සංසන්දනය කරයි.

මෙම අගයන් සමානවුවහොත් එම case statement හි සඳහන් කාර්යය සිදුකරයි. මෙම switch Statement කොටස අවසානයේ break විධානය දැකිය හැකිය. මෙම විධානය මගින් මෙම ප්‍රෝග්‍රෑම කොටස නැවතීමට අවශ්‍ය විධානය ලබාදෙයි.

switch...case controls the flow of programms by allowing programmers to specify different code that should be executed in various conditions.
In particular, a switch statement compares the value of a variable to the values specified in case statements. When a case statement is found whose value matches that of the variable, the code in that case statement is run.

The break keyword exits the switch statement, and is typically used at the end of each case. Without a break statement..

උදාහරණ Example :-

```
switch (var) {
case 1:
  //do something when var equals 1
  break;
case 2:
  //do something when var equals 2
  break;
default:
  // if nothing else matches, do the default
  // default is optional
```

char data type

char සඳහා හඳුන්වන්නේ 1 බයිටයක් අඩංගු data වර්ගයක් වන අතර මෙහි -128 සිට 127. දක්වා අංක හෝ අක්ෂර ගබඩාත් කළහැකිය.

A data type that takes up 1 byte of memory that stores a character value. Character literals are written in single quotes, like this: 'A' (for multiple characters - strings - use double quotes: "ABC").
The char datatype is a signed type, meaning that it encodes numbers from -128 to 127. For an unsigned, one-byte (8 bit) data type.

උදාහරණ Example :-

```
char myChar = 'A';  - 1 byte( 8bits)
char myChar = 65;
```

unsigned char

මෙවැනි data වර්ගයක අංක 0 සිට 255 දක්වා වූ අගයන් ගබඩාකර තැබිය හැකිය.

An unsigned data type that occupies 1 byte of memory. Same as the byte datatype.
The unsigned char datatype encodes numbers from 0 to 255.

උදාහරණ Example :-

```
unsigned char myChar = 240;
```

Arrays

Array යනු විචල්‍ය අගයන්ගේ එකතුවක් වන අතර ඒවා අංක මගින් තෝරාගතහැකිය.
is a collection of variables that are accessed with an index number.

Creating (Declaring) an Array

Array හඳුන්වාදීමට පහත දැක්වෙන ක්‍රම තෝරා ගතහැකිය

උදාහරණ Example :-

```
int myInts[6];
int myPins[] = {2, 4, 8, 3, 6};
char message[6] = "hello";
```

string

ප්‍රෝග්‍රෑමයක් මගින් යම් අක්ෂර මාලවාක් LCD පුවරුවක හෝ පරිගණකයේ දැක්වීමට අවශ්‍ය වූ විට එම අක්ෂර මාලව පහත දැක්වෙන ආකාරයට ප්‍රෝග්‍රෑමයේ ලිවියයුතුය.

Text strings can be represented in two ways. you can use the String data type

උදාහරණ Example :-

```
char Str1[15];
char Str2[8] = {'a', 'r', 'd', 'u', 'i', 'n', 'o'};
char Str3[8] = {'a', 'r', 'd', 'u', 'i', 'n', 'o', '\0'};
char Str4[ ] = "arduino";
char Str5[8] = "arduino";
char Str6[15] = "arduino";
```

දීර්ඝ අක්ෂර මාලා පහත සැක්වෙන අකාරයට සැකසිය හැකිය

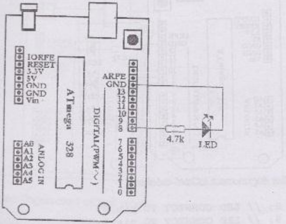
```
char myString[] = "This is the first line"
" this is the second line"
" this is the thierd line ";
```

උදාහරණ Example :-

```
char* myStrings[]={"This is string 1", "This is string 2",
"This is string 3","This is string 4", "This is string 5",
"This is string 6"};
```

Project 01
පරීක්ෂණ අංක 01

මෙම පරීක්ෂණයේදී පින් අංක 8 ට සම්බන්ධ කළ LED බල්බයක් ස්වයංක්‍රීයව ON හා OFF කරන ආකාරය සොයා බලමු.
ඒ සඳහා පහත දැක්වෙන පරිපථය සකසන්න.



පහත දැක්වෙන ප්‍රෝග්‍රෑමය මෘදුකාංගයේ සටහන් කරන්න.

```
// Program 1: Making an LED blink on and off

int lED = 8; // LED CONNECT TO pin 8

void setup()
{
pinMode(LED, OUTPUT); // tell Arduino the pin 10 IS an output
}
void loop()
{
  digitalWrite(LED, HIGH);      // LED  ON
  delay(1000);                         // wait ONE second DELAY
  digitalWrite(LED, LOW);       // LED OFF
  delay(1000);                         // wait ONE second DELAY
}
```

පළමුව මෙම ප්‍රෝග්‍රෑමය හෝ sketch ය ෆෝල්ඩරයක් තුළ save කරන්න.

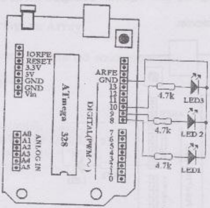
දැන් මෘදුකාංගය ආධාරයෙන් මෙම ප්‍රෝග්‍රෑමය හෝ sketch ය verify/conpile කරන්න.

නිවැරදිව ලියා ඇත්නම් මෘදුකාංගයේ පහල කොටසේ එය Done compiling ලෙස දැක්වේ.
දැන් මෙය Ardiuno පරිපථයට Upload කරන්න.

මෙම නිර්මානයට අදාල sketch ය CD තැටියේ LED ON OFF ලෙස දක්වා ඇති Folder තුළ ලබාදී ඇත.

පරීක්ෂණ අංක 02

මෙම පරීක්ෂනයේදී පින් අගු 8,9,10 ට ඈදුම්කල LED බල්බ 3 ක් ස්වයංක්‍රියව ON හා OFF කරන ආකාරය සොයා බලමු.
ඒ සඳහා පහත දැක්වෙන පරිපථය සකසන්න.



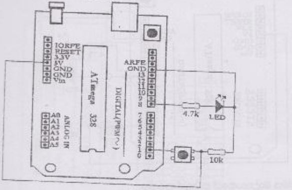
පහත දැක්වෙන ප්‍රෝග්‍රෑමය මාදුකාංගයේ සටහන් කරන්න.

```
int LED1 = 8; // LED CONNECT TO pin 8
int LED2 = 9; // LED CONNECT TO pin 9
int LED3 = 10;// LED CONNECT TO pin 10

void setup()
{
pinMode(LED1, OUTPUT); // pin 8,9,10 IS an output
pinMode(LED2, OUTPUT);
pinMode(LED3, OUTPUT);
}
void loop()
{
  digitalWrite(LED1, HIGH);        // LED  ON
  delay(1000);                        // wait ONE second DELAY
  digitalWrite(LED2, HIGH);        // LED ON
  delay(1000);                        // wait ONE second DELAY
  digitalWrite(LED3, HIGH);        // LED ON
  delay(1000);                        // wait ONE second DELAY

  digitalWrite(LED3, LOW);         // LED OFF
  delay(1000);
  digitalWrite(LED2, LOW);         // LED OFF
  delay(1000);
  digitalWrite(LED1, LOW);         // LED OFF
  delay(1000);
  }
```

මෙම නිර්මානයට අදාල sketch ය CD තැටියේ LED1,2,3 ON OFF ලෙස දක්වා ඇති Folder තුළ ලබාදී ඇත.

Project 03
පරීක්ෂණ අංක 03

මෙම පරීක්ෂණයේදී පින් අග්‍ර 1 ට, ඇඳුම්කර ඇති බටනයක් එබීමෙන් පින් අග්‍ර 8 ට සම්බන්ධ LED බල්බයක් ON කරන ආකාරය සොයා බලමු.
ඒ සඳහා පහත දැක්වෙන පරිපථය සකසන්න.



පහත දැක්වෙන ප්‍රෝග්‍රෑමය මාදුකාංගයේ සටහන් කරන්න.

```
const int button = 1;//pushbutton pin
const int LED =  8; //LED pin
int buttonState = 0; // pushbutton status

void setup() {

  pinMode(LED, OUTPUT);
  pinMode(button, INPUT);
}
void loop() {
  // read the state of the pushbutton value:
  buttonState = digitalRead(button);
   // check if the pushbutton is pressed.
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(LED, HIGH);
  }
  else { // එබී නැතිනම් led බල්බය OFF කරයි
    // turn LED off:
    digitalWrite(LED, LOW);
  }}
```

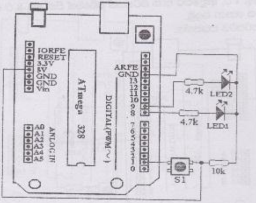
buttonState = 0 මගින් පළමුව බටනය එබී නැතිබව දන්වයි. if විධානය කොටසින් බටනය එබී තිබේදැයි විමසා බලයි. එසේ එබී ඇතිනම් (if (buttonState == HIGH)) පමණක් LED1 බල්බය ON කළයුතු බව දන්වා පහත ප්‍රෝග්‍රෑම කොටසින් දන්වයි.

```
if (buttonState == HIGH) { බල්බය එබී ඇතිනම්
    // turn LED on:
    digitalWrite(LED, HIGH);
  }
```

මෙම පරීක්ෂණයට අදාල sketch ය CD තැටියේ BUTTON ලෙස දක්වා ඇති Folder තුල ලබාදී ඇත.

පරීක්ෂණ අංක 04

මෙම පරීක්ෂණයේදී පින් අග්‍ර 1 ට ඇඳුම්කර ඇති බටනයක් එබී ඇතිනම් පින් අග්‍ර 8, 9 ට සම්බන්ධ LED බල්බ මිලි තත්පර 500 ක වේගයෙන් ON සහ OFF වීමට සැලැස්වීමට හැකි ප්‍රෝග්‍රෑමයක් සකසමු.
මේ සඳහා පහත දැක්වෙන පරිපථය සකසන්න.



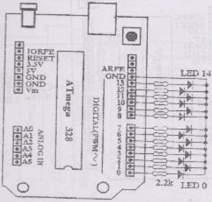
පහත දැක්වෙන ප්‍රෝග්‍රෑමය මෘදුකාංගයේ සටහන් කරන්න.

```
int LED1 = 8; // setup initializes serial and the led pin
int LED2 = 9; // setup initializes serial and the led pin
int button = 1;
int buttonState = 0;
void setup()
{
  pinMode(LED1, OUTPUT); //
  pinMode(LED2, OUTPUT); //
  pinMode(button,INPUT); //
}
void loop()
{
  // read the state of the pushbutton value:
  buttonState = digitalRead(button);
if (buttonState == HIGH) // if button is pressed
 {//   if විධානය කොටසින් බටනය එබී ඇතිදැයි විමසා බලයි.

   digitalWrite(LED1, HIGH); // turn on led1
delay(500);
   digitalWrite(LED2, HIGH); // turn on led3
delay(500);
}// එසේ එබී ඇතිනම් පමනක් LED1,LED2 බල  ON/OFF කලයුතු බව දන්වයි
else(buttonState == LOW)
{ digitalWrite(LED1, LOW); // turn off led1,2
} }//එසේ නොමැතිනම් LED1,2 බල්බ OFF කරන ලෙසටත් පහත ප්‍රෝග්‍රෑම කොටසින් දන්වයි.
```

මෙම නිර්මානයට අදාල sketch ය CD තැටියේ IF ELSE ලෙස දක්වා ඇති Folder තුල ලබාදී ඇත.

Project 05
පරීක්ෂණ අංක 05

මෙම පරීක්ෂණයේදී පින් අගු 0 සිට 13 දක්වා ඇඳුම්කර ඇති LED බල්බය පිලිවෙලින් ON කිරීමත් ඒවා නැවතත් OFF කරන ආකාරයක් සෙවීමට ප්‍රෝග්‍රෑමයක් සකසමු.
මේ සඳහා පහත දැක්වෙන පරිපථය සකසන්න.



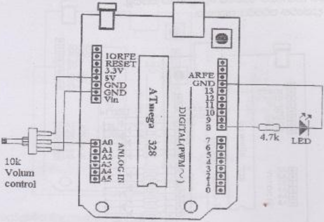
පහත දැක්වෙන ප්‍රෝග්‍රෑමය මාදුකාංගයේ ටයිපන් කරන්න.

```
int timer = 100;  // The higher the number, the slower the timing.
int ledPins[] = {0,1, 2, 3, 4, 5, 6,7,8,9,10,11,12,13};
int pinCount = 14;    // the number of leds connect to pii 14
void setup() {

  for (int thisPin = 0; thisPin < pinCount; thisPin++) {
  pinMode(ledPins[thisPin], OUTPUT);
  }
}
void loop() {

//මෙම කොටසින් LED 0 සිට 14 දක්වා පිළිවෙලින් ON කරයි
 for (int thisPin = 0; thisPin < pinCount; thisPin++) {

    digitalWrite(ledPins[thisPin], HIGH);
    delay(timer);

    digitalWrite(ledPins[thisPin], LOW);
  }
//මෙම කොටසින් ON වී ඇති පිළිවෙලින් OFF කරයි

  for (int thisPin = pinCount - 1; thisPin >= 0; thisPin--) {

    digitalWrite(ledPins[thisPin], HIGH);
    delay(timer);
    digitalWrite(ledPins[thisPin], LOW);
```

මෙම නිර්මාණයට අදාළ sketch ය CD තැටියේ LED14 ON OFF ලෙස දක්වා ඇති Folder තුළ ලබාදී ඇත.

පරීක්ෂණ අංක 06

මෙම පරීක්ෂණයේදි ඇනලොග් පින් අග්‍ර 0 ට ඇඳුම්කර ඇති පරිමා පාලකයක් මගින් Digital පින් අග්‍ර 8ට සවිකල LED බල්බයක ආලෝක තීව්‍රතාවය වෙනස් කරන ආකාරය සොයා බලමු.
මේ සඳහා පහත දැක්වෙන පරිපථය සකසන්න.



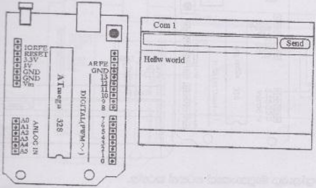
පහත දැක්වෙන ප්‍රෝග්‍රැමය මාදුකාංගයේ සටහන් කරන්න.

```
int sensorPin = A0;    // පරිමා පාලකය anlog පින් අග්‍ර 0 ට ඇඳුම්කර ඇත
int ledPin = 8;        // LED connect to gigita pin 8
int sensorValue = 0; // variable to store the value coming from the
sensor

void setup() {
  pinMode(ledPin, OUTPUT); // ledPin as an OUTPUT
}
void loop() {

     sensorValue = analogRead(sensorPin); // පරිමා පාලකය සම්බන්ධ පින් අග්‍රයේ
            වටිනාකම කියවයිread the value from the sensor

  digitalWrite(ledPin, HIGH); // turn the ledPin on

  delay(sensorValue); // ඩිලේ අගය සඳහා ඉහත  A0 අග්‍රය කියවා ලබාගත් අගය
     ඩිලේ හි (sensorValue) අගය වෙත ආදේශකර එම කාලය නැවතී සිටී // turn the
     ledPin off:

  digitalWrite(ledPin, LOW);
  //
     delay(sensorValue); ඩිලේ අගය සඳහා ඉහත  A0 අග්‍රය කියවා ලබාගත් අගය    ඩිලේ
     හි (sensorValue) අගය වෙත ආදේශකර එම කාලය නැවතී සිටී // turn the
}
```

මෙම නිර්මානයට අදාල sketch ය CD තැටියේ Anlog ලෙස දක්වා ඇති Folder තුල ලබාදී ඇත.

Project 07
පරීක්ෂණ අංක 07

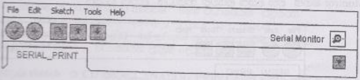
මෙම පරීක්ෂණයේදී ඇනලොග් පින් අංක 0 ට ඈඳුම්කර ඇති සංවේදී උපාංගය (sensor) මගින් ලැබෙන දත්ත Arduino මාදුකාංගයේ ඇති serial monitor Window වේ දර්ශනය කරගන්නා ආකාරය සොයා බලමු. මේ සඳහා පහත දැක්වෙන පරිපථය සකසන්න.



පහත දැක්වෙන ප්‍රෝග්‍රෑමය මාදුකාංගයේ සටහන් කරන්න.

```
// Program 7: Make Arduino Print "Hello World" to Computer

        void setup() {
        Serial.begin(9600); // open a 9600 baud communication line to
                                      computer
        }
        void loop()
        {
        Serial.println("Hello"); // write the word "Hello"
        delay(1000);             // wait a second
        Serial.println("World"); // write the word "World"
        delay(1000);             // wait a second
        }
```

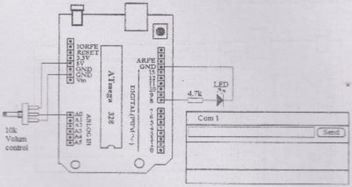
Arduino මාදුකාංගයේ ඇති serial monitor Window ලබා ගැනීම සඳහා මාදුකාංගයේ පහත පෙන්වා ඇති බටනය එබීමෙන් එය Open කරගතහැකිය. ප්‍රෝග්‍රෑමය Ardiuno පරිපථයට upload කරන්න. Serila Monitor මගින් එහි දත්ත ලියවෙන ආකාරය දැකිය හැකිය.



මෙම නිර්මාණයට අදාල sketch ය CD තැටියේ SERIAL PRINT ලෙස දක්වා ඇති Folder තුල ලබාදී ඇත.

පරීක්ෂණ අංක 08

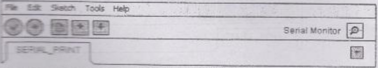
මෙම පරීක්ෂණයේදී ඇනලොග් පින් අග්‍ර 0 ට ඈඳුම්කර ඇති පරිමා පාලකයක් දෙපසට කැරකැවීමෙන් වෙනස්වන වෝල්ටියතාවයට අනුව Digital පින් අග්‍ර 8 ට සවිකල LED බල්බයක තීව්‍රතාවය හා එම වෝල්ටියතා අගය Arduino මෘදුකාංගයේ ඇති serial monitor Window වේ දර්ශනය කරගන්නා ආකාරය සොයා බලමු.
පහත දැක්වෙන පරිපථය සකසන්න.



පහත දැක්වෙන ප්‍රෝග්‍රැමය මෘදුකාංගයේ සටහන් කරන්න.

```
void loop() {
 // read the analog in value:
 sensorValue = analogRead(analogInPin);

 // map it to the range of the analog out:

 outputValue = map(sensorValue, 0, 1023, 0, 255);

 // change the analog out value:
 analogWrite(analogOutPin, outputValue);

 // print the results to the serial monitor:

 Serial.print("sensor = " );
 Serial.print(sensorValue);
 Serial.print("\t output = ");
 Serial.println(outputValue);
```

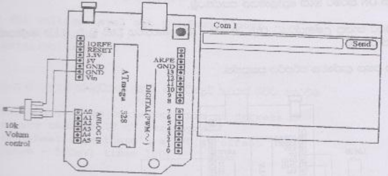
Arduino මෘදුකාංගයේ ඇති serial monitor Window ලබා ගැනීම සඳහා මෘදුකාංගයේ පහත පෙන්වා ඇති බටනය එබීමෙන් එය Open කරගතහැකිය. ප්‍රෝග්‍රැමය Ardiuno පරිපථයට upload කරන්න. Serila Monitor මගින් එහි දත්ත ලියවන ආකාරය දැකිය හැකිය.



මෙම නිර්මාණයට අදාළ සංකේත ය CD තැටියේ ANALOG LED SERIAL ලෙස දක්වා ඇති Folder තුළ ලබාදී ඇත.

Project 09
පරීක්ෂණ අංක 09

මෙම පරීක්ෂණයේදී ඇනලොග් පින් අංක 0 ට ඈඳුම්කර ඇති පරිමා පාලකයක් දෙපසට කැරකැවීමෙන් 0V සිට 5V,DC විභවයේ වෙනස්වීම Arduino මෘදුකාංගයේ serial monitor window මේ දර්ශනය කරගන්නා ආකාරය සොයා බලමු.
මේ සඳහා පහත දැක්වෙන පරිපථය සකසන්න.



පහත දැක්වෙන ප්‍රෝග්‍රැමය මෘදුකාංගයේ සටහන් කරන්න.

```
int sensorValue = 0

 void setup() {

  Serial.begin(9600); // initialize serial communication at
                      9600 bits per second:

  }
 void loop() {
  int sensorValue = analogRead(A0); //read the input pin 0:
   // Convert the analog reading (which goes from 0 - 1023)
  to a voltage (0 - 5V):
  float voltage = sensorValue * (5.0 / 1023.0);

  Serial.println(voltage); // print value on serialmonitor:
}
```

පළමුව පරිගණකය හා Arduino මෘදුකාංගයයේ ඇති serial monitor Window අතර දත්ත හුවමාරුව සඳහා baud rate අගය මෙසේ සකසමු `Serial.begin(9600)`
මෙම අගය serial monitor වින්ඩොවේ පහලින් සැකසිය හැකිය.
`int sensorValue = analogRead(A0);` පරිමා පාලකය මගින් A0 ට ලබදෙන ඇනලොග් වෝල්ටීයතා (0 සිට 5V) අගය `int sensorValue = analogRead(A0);` මගින් කියවයි.

මෙම වෝල්ටීයතාවය ඇනලොග් අගයන් බවට හැරවීම මෙම විධානයෙන් සිදුකරයි
`float voltage = sensorValue * (5.0 / 1023.0);` හරවයි.
මෙම අගය `Serial.println(voltage)` යේ සටහන් කරන ලෙස මෙම
`Serial.println(voltage);` විධානයෙන් ඇනුම් දෙයි

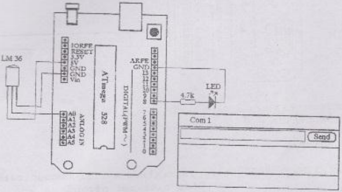
මෙම පරීක්ෂණයට අදාල sketch ය CD තැටියේ **ANALOG PRINT** ලෙස දක්වා ඇති **Folder** තුල ලබාදී ඇත.

Project 10
පරීක්ෂණ අංක 10

මෙම පරීක්ෂණයේදී ඇනලොග් පින් අඟු LM35 / 36 උෂ්ණත්වමාන sensor උපාංගයක් ඇඳුම්කර යම් පෙර සැකසූ උෂ්ණත්ව අගයක් ඉක්මවා උෂ්ණත්වය ඉහළ ගියවිට එම අගය Arduino මෘදුකාංගයේ ඇති serial monitor window වේ දර්ශනය වීමටත් පරිපථයට ඇඳුම් කර ඇති LED බල්බයක් හෝ රිලේ උපාංගයක් ON කිරීමට හැකි ප්‍රෝග්‍රෑමයක් සකස්කරමු.

මෙහිදී පෙර සැකසූ උෂ්ණත්වයට වඩා අංශක 1 ක් උෂ්ණත්වය වැඩි වූ ට LED බල්බයක් හෝ රිලේ උපාංගයක් ON වේ.

මේ සඳහා පහත දැක්වෙන පරිපථය සකසන්න.



පහත දැක්වෙන ප්‍රෝග්‍රෑමය මෘදුකාංගයේ සටහන් කරන්න.

```
const int sensorPin = A0;//LM35 / LM36  ඇඳුම්කර ඇති පින් අඟුය
const float baselineTemp = 20.0;// පෙර සැකසූ උෂ්ණත්ව අගය
int LED = 8; // int relay = 8;//LED connect to pin 8

void setup()
{
  Serial.begin(9600);

   pinMode(LED, OUTPUT);
   digitalWrite(LED, LOW);
  }

void loop()
{
  // read the value on AnalogIn pin 0
  // and store it in a variable SensorVal
  int sensorVal = analogRead(sensorPin);
```

Arduino මෘදුකාංගයේ ඇති serial monitor window වේ පින් අඟු 0 ට අදාළ අගය දර්ශනය වීමට

```
  Serial.print("sensor Value: ");
  Serial.print(sensorVal);
```

ලබාගත් LM36 ය sensor යේ වෝල්ටියතා අගය ඩිජිටල් අගයක් බවට හැරවීම

```
// convert the ADC reading to voltage
float voltage = (sensorVal / 1024.0) * 5.0;
```

මෙම අගය serial monitor window වේ දර්ශනය වීමට

```
// Send the voltage level out the Serial port
Serial.print(", Volts: ");
Serial.print(voltage);
```

ලබාගත් LM36 ය sensor යේ වෝල්ටියතා අගය සෙන්ටිග්‍රේඩ් අගයක් බවට හැරවීම

```
// convert the voltage to temperature in degrees C
// the sensor changes 10 mV per degree
// the datasheet says there's a 500 mV offset
// ((volatge - 500mV) times 100)

Serial.print(", degrees C: ");
float temperature = (voltage - .5) * 100;
```

ලබාගත් වෝල්ටියතා අගයෙන් 0.5ක් අඩු කිරීමත් එම අගය 100 න් ගුණකිරීමක් මෙහිදී සිදුකරයි

```
//(voltage - .5) * 100;
```

මෙම අගය serial monitor window වේ දර්ශනය වීමට

```
Serial.println(temperature);
```

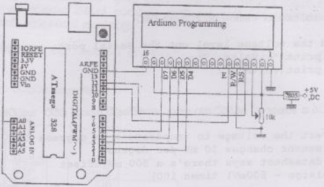
LM35 sensor යෙන් ලැබෙන උෂ්ණත්වය හා පෙර සැකසූ උෂ්ණත්වය අගය සන්සන්දනය කර බලා එම අගය අංශක 1 කින් ඉහල ගොස් ඇතිනම් LED බල්බය ON කිරීමටත් එසේ නොවේනම් LED බල්බය OFF කිරීමටත් උපදෙස් ලබාදෙයි.

```
  // if the current temperature is lower than the pre-set
temperture turn off all LED

if (temperature >= baselineTemp + 1) // අංශක 1 කින් ඉක්මවා ඇතිදැයි බැලීම
{
    digitalWrite(LED, HIGH); // LED බල්බය ON කිරීම
  }

else if (temperature < baselineTemp)//  අංශක 1 කින් ඉක්මවා නැතිනම්
{
    digitalWrite(LED, LOW); LED බල්බය OFF කිරීම

    delay(20); delay  for 20 ms
}}}
```

මෙම නිර්මානයට අදාල sketch ය CD තැටියේ LM35 TEMP CONTROL ලෙස දක්වා ඇති Folder තුල ලබාදී ඇත.

Project 11
පරීක්ෂණ අංක 11

LCD පුවරුවක් Ardiuno පරිපථය හා සම්බන්ධ කරගන්නා ආකාරය හා LCD පුවරුව මත අක්ෂර,අංක හෝ වැකි සටහන් කරගන්නා ආකාරය සොයා බලමු.
මේ සඳහා පහත දැක්වෙන පරිපථය සකසන්න.



පහත දැක්වෙන ප්‍රෝග්‍රෑමය මෘදුකාංගයේ සටහන් කර LCD පුවරුවේ ("Arduino Programming ") ලෙස සටහන් කරගන්නා ආකාරය සොයා බලමු

#include <LiquidCrystal.h> //මෙමගින් LCD ,Library ය වෙත මෘදුකාංගය ගෙනයයි.

LiquidCrystal lcd(12,11,5,4,3,2); // LCD පුවරුවේ ඇති
RS =12 ,E=11, D4=5, D5=4, D6=3, D7=2
යන අග්‍ර Ardiuno පරිපථයේ ඈඳුම් වී ඇති පින් අග්‍ර මෘදුකාංගයට දැනුම් දෙයි.

void setup()
{
lcd.begin(16,2); LCD පුවරුවේ 16 x 2 වර්ගයේ උපාංගයක් බව දැනුම් දෙයි.
}

void loop()
{
lcd.clear(); // LCD පුවරුව යමක් සඳහන් වී ඇතිනම් එය ඉවත් කරන ලෙස දැනුම් දෙයි.
lcd.setCursor(0,0); // කසරය LCD පුවරුවේ මුලට ගෙනයයි.
lcd.print("Arduino Programming "); // මෙම වැකිය LCD පුවරුවේ සඳහන් කරන ලෙස දැනුම් දෙයි

delay(1000); විනාඩි 1 ක කාලයක් මෙම වැකිය LCD පුවරුවේ තබාගන්නා ලෙසත් කාලය අවසානයේ එය LCD පුවරුවෙන් එම වැකිය ඉවත් කිරීමටත් දැනුම් දෙයි.

lcd.clear();
lcd.print("Good Morning"); // මෙම වැකිය LCD පුවරුවේ සඳහන් කරන ලෙස දැනුම් දෙයි.
delay(1000); විනාඩි 1 ක කාලයක් මෙම වැකිය LCD පුවරුවේ තබාගන්නා ලෙසත් කාලය අවසානයේ එය LCD පුවරුවෙන් එම වැකිය ඉවත් කිරීමට හා නැවතත් ආරම්භය වෙත යන ලෙස උපදෙස් දෙයි.

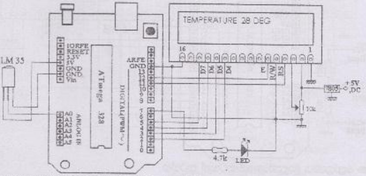
} }
මෙම නිර්මානයට අදාළ sketch ය CD තැටියේ PROJECTS ලෙස දක්වා ඇති Folder යේ LCD DISPLAY තුල ලබාදී ඇත.

පරීක්ෂණ අංක 12

මෙම පරීක්ෂණයේදී ඇනලොග් පින් අග්‍ර LM35 ය උෂ්ණත්වමාන sensor උපාංගයක් ඇඳුම්කර යම් පෙර සැකසු උෂ්ණත්ව අගයක් ඉක්මවා උෂ්ණත්වය ඉහල ගියවිට ඒබ ව LCD (16 x 2) පුවරුවක දර්ශනය වීමටත් එවිට LED බල්බයක හෝ රිලේ උපාංගයක් ON වීමටත් සැලැස්විය හැකි ප්‍රෝග්‍රෑමයක් සකස් කරමු.

මෙහිදී පෙර සැකසු උෂ්ණත්ව අංශක 35 වේ. එම අගය වඩා අංශක 1 ක් උෂ්ණත්වය වැඩි වු විට LED බල්බයක් හෝ රිලේ උපාංගයක් ON විමට සලස්වි ඇත.

පහත දැක්වෙන පරිපථය සකසන්න.



පහත දැක්වෙන ප්‍රෝග්‍රෑමය මාදුකාංගයේ සටහන් කරන්න.

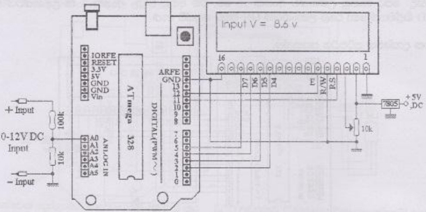
```
#include <LiquidCrystal.h>// LCD ( 16 x 2) පුවරුවක library වෙත ගෙනයයි
int reading = 0;
int sensorPin = A0; //LM36  සම්බන්ද කල පින් අග්‍රය
int relay =7; // රිලේ උපාංගය සවිවන පින් අග්‍රය

LiquidCrystal lcd(11, 12, 5, 4, 3, 2);//LCD පුවරුවේ පින් අග්‍ර සවිවන
ආකාරය
void setup() {
lcd.begin(16, 2); lcd පුවරුවේ වර්ගය
pinMode(relay,OUTPUT); රිලේ උපාංගය ක්‍රියා කරනුයේ output ලෙස බව දැනුම් දෙයි
}
void loop() {
        reading = analogRead(sensorPin); //LM35 කියවීම
        int celsius = reading/2; // කියවු අගය 2 න් බෙදීම
        lcd.setCursor(0, 0);
        lcd.print("Temperature: ");
        lcd.setCursor(0,1);
        lcd.print(celsius, DEC);
        lcd.print("C");
if (celsius >35) {//උෂ්ණත්වය අංශක 35 වඩා ඉහල නම් රිලේ උපාංග ON කිරීම
digitalWrite(7,LOW);
            digitalWrite(7,HIGH);
            } else { // එසේ නැතිනම් රිලේ උපාංලය OFF කිරීමට උපදෙස් දීම
            digitalWrite(7,LOW);}
            delay(500); ඩිලේ කාලය මිලි තත්පර 500 ලෙස සැකසීම
            lcd.clear(); }//LCD පුවරු clear කිරීම
```

මෙම නිර්මානයට අදාල sketch ය CD තැටියේ LCD LM35 TEMP CONREOL ලෙස දක්වා ඇති Folder තුල ලබාදී ඇත.

පරීක්ෂණ අංක 13

මෙම පරීක්ෂණයේදී ඇනලොග් පින් අග්‍ර 0 මගින් පරිපථයට ලැබෙන වෝල්ටීයතාවය ගණනයකර එම අගය LCD (16 x 2) පුවරුවක පෙන්වීමට හැකි ප්‍රෝග්‍රෑමයක් සකස්කරන ආකාරය සොයා බලමු.

මේ සඳහා පහත දැක්වෙන පරිපථය සකසන්න.



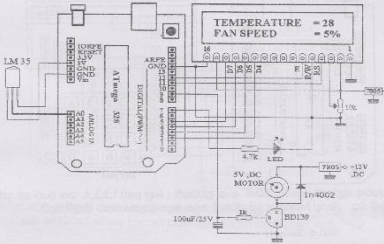
පහත දැක්වෙන ප්‍රෝග්‍රෑමය මාදුකාංගයේ සටහන් කරන්න.

```
include <LiquidCrystal.h>
LiquidCrystal lcd(11,12,5,4,3,2);
int analogInput = 0;
  float vout = 0.0;
float vin = 0.0;
float R1 = 100000.0; // resistance of 100K
float R2 = 10000.0; // resistance of 10K
int value = 0;
void setup(){
   pinMode(analogInput, INPUT);
   lcd.begin(16, 2);
   lcd.print("DC VOLTMETER");
}
void loop(){
   // read the value at analog input
   value = analogRead(analogInput);
   vout = (value * 5.0) / 1024.0;

   vin = vout / (R2/(R1+R2));
   if (vin<0.09)
  {
   vin=0.0
}
lcd.setCursor(0, 1);
lcd.print("INPUT V= ");
lcd.print(vin);
delay(500);
```

මෙම නිර්මානයට අදාල sketch ය CD තැටියේ VOLT METER ලෙස දක්වා ඇති Folder තුල ලබාදී ඇත.

Project 14
පරීක්ෂණ අංක 14

මෙම පරීක්ෂණයේදී ඇනලොග් පින් අංක 0 ට ඇදුම්කල LM35 උපාංගයක් මගින් උෂ්ණත්වය මැනීමටත් එම උෂ්ණත්ව අගයට අනුව පරිපථයට සවිකර ඇති DC මෝටරයක භ්‍රමණ වේගය පාලනය කලහැකි ප්‍රෝග්‍රෑමයක් සකස්කරන ආකාරය සොයා බලමු.



පහත දැක්වෙන ප්‍රෝග්‍රෑමය මෘදුකාංගයේ සටහන් කරන්න.

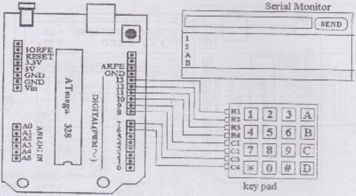
```
        #include <LiquidCrystal.h>//source:olled-temperature/
LiquidCrystal lcd(11,12,5,4,3,2); //library initialization
int tempPin = A0; // the output pin of LM35

int fan = 9; // the pin where fan is
int led = 7; // led pin
int temp;
int tempMin = 35; // the temperature to start the fan
int tempMax = 50; // the maximum temperature when fan is at 100%
int fanSpeed;
int fanLCD;
int ldrLCD;
lcd.print("TEMP: ");
lcd.print(temp); // display the temperature
lcd.print("C ");
//lcd.print("LIGHT");
lcd.setCursor(0,1); // move cursor to next line
//lcd.print("FAN: ");
lcd.print(fanLCD); // display the fan speed
 lcd.print("%");
lcd.print(sensorValue);
delay(200);
lcd.clear();
}return temp * 0.48828125; }
```

මෙම නිර්මාණයට අදාල sketch ය CD තැටියේ TEMP FAN CONTROL ලෙස දක්වා ඇති Folder තුල ලබාදී ඇත.

පරීක්ෂණ අංක 15

මෙම පරීක්ෂණයේදී Ardiuno Uno පරිපථයකට ඇදුම්කල KEY PAD යක් ආධාරයෙන් අංක Serial Monitor යේ සටහන් කරගන්නා ආකාරය සොයා බලමු.

මේ සඳහා පහත දැක්වෙන පරිපථය සකසන්න.



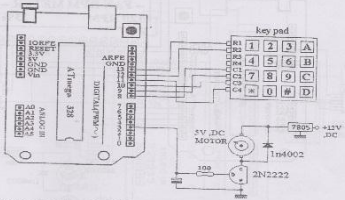
පහත දැක්වෙන ප්‍රෝග්‍රෑම් කොටස මගින් අංක පුවරුවේ (Key pad) 1,2,3,A යන අංක හා අක්ෂරය එබී ඇතිනම් එය Serial Moniotor යේ සටහන් කරන ලෙස මෘදුකාංගයට දැනුම් දෙයි.

```
void loop()
{
 digitalWrite(r1,LOW);
 digitalWrite(r2,HIGH);
 digitalWrite(r3,HIGH);
 digitalWrite(r4,HIGH);
 colm1=digitalRead(c1);
 colm2=digitalRead(c2);
 colm3=digitalRead(c3);
 colm4=digitalRead(c4);
 if(colm1==LOW)
 {Serial.println("1");
  delay(200);}
 else{
  if(colm2==LOW)
  {Serial.println("2");
   delay(200);}
  else{
  if(colm3==LOW)
  {Serial.println("3");
   delay(200);}
  else{
  if(colm4==LOW)
  {Serial.println("A");
     delay(200);
}}}}}
```

මෙම නිර්මානයට අදාල sketch ය CD තැටියේ KEY PAD DISPLAY ලෙස දක්වා ඇති Folder තුල ලබාදී ඇත.

Project 16
පරීක්ෂණ අංක 16

මෙම පරීක්ෂණයේදී Ardiuno Uno පරිපථයකට ඈදුම්කල KEY PAD යක් ආධාරයෙන් DC මෝටරයක වේගය පාලනය කරගන්නා ආකාරය සොයා බලමු.
මේ සඳහාපහත දැක්වෙන පරිපථය සකසන්න.



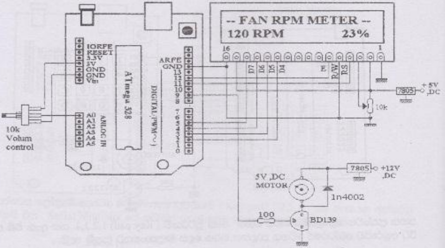
පහත දැක්වෙන ප්‍රෝග්‍රෑම් කොටස මගින් අංක පුවරුවේ (Key pad) 1,2,3,4 යන අංක එබ් ඇතිනම් එය ඊට අනුරූපීව මෝටරයේ වේගය පාලනය කරන ලෙස මිඳුකාංගයට දැනුම් දෙයි.

```
        void loop()
{
  digitalWrite(r1,LOW);
  digitalWrite(r2,HIGH);
  colm1=digitalRead(c1);
  colm2=digitalRead(c2);
  colm3=digitalRead(c3);
  colm4=digitalRead(c4);
  if(colm1==LOW)            //checks whether key "1" is pressed.
  { analogWrite(pwm,42); // writes "42" (duty cycle 16%).
   delay(200);}
  else
  {
   if(colm2==LOW)           //checks whether key "2" is pressed.
   { analogWrite(pwm,84); // writes "84" (duty cycle 32%).
    delay(200);}
   else
   {
   if(colm3==LOW)           //checks whether key "3" is pressed
   {analogWrite(pwm,126); // writes "126" (duty cycle 48%).
   delay(200);}
   else
   {
   if(colm4==LOW)           // checks whether key"A" is pressed.
  {digitalWrite(pwm,LOW);//   (duty cycle 0%).MotorOFF.
   delay(200);}

            }}}}}
```

මෙම නිර්මාණයට අදාල sketch ය CD තැටියේ KEY PAD MOTOR SPEED ලෙස දක්වා ඇති Folder තුල ලබාදී ඇත.

පරීක්ෂණ අංක 17

මෙම පරීක්ෂණයේදී ඇනලොග් පින් අග්‍ර 0 ට අඳුම්කල 10k පරිමා පාලකයක් ආධාරයෙන් DC මෝටරයක භ්‍රමණ වේගය පාලනය කිරීමත් එම වේගය LCD පුවරුවේ පෙන්නුම් කරගන්නා ආකාරයත් සොයා බලමු.



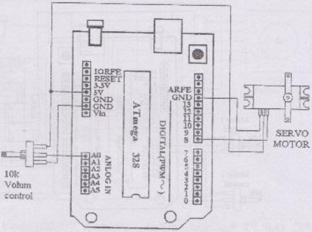
පහත දැක්වෙන ප්‍රෝග්‍රැමය මෘදුකාංගයේ සටහන් කරන්න.

```
void loop()
{
delay(1000);
detachInterrupt(0);          //detaches the interrupt
time=millis()-oldtime;       //finds the time
rpm=(rev/time)*60000;        //calculates rpm
oldtime=millis();            //saves the current time
rev=0;
value=analogRead(pot);       //reads the speed control POT
value=value/4;
analogWrite(pwm,value);      //sets the desired speed
percent=(value/255)*100;     //finds the duty cycle %
lcd.clear();
lcd.setCursor(0,0);
lcd.print("_FAN RPM METER_");
lcd.setCursor(0,1);
lcd.print(rpm);
lcd.print(" RPM");
lcd.print("  ");
lcd.print(percent);
lcd.print("%");
attachInterrupt(0,isr,RISING);
}
```

මෙම නිර්මාණයට අදාල sketch ය CD තැටියේ FAN RPM METER ලෙස දක්වා ඇති Folder තුළ ලබාදී ඇත.

Project 18
පරීක්ෂණ අංක 18

Ardiuno පරිපථයක් භාවිතයෙන් SERVO MOTOR යක් 10k පරිමා පාලකයන් ආධාරයෙන් දෙපසට භ්‍රමණය කරන ආකාරය සොයා බලමු.
මේ සඳහා පහත දැක්වෙන පරිපථය සකසන්න.



පහත දැක්වෙන ප්‍රෝග්‍රැමය මාදූකාංගයේ සටහන් කරන්න.

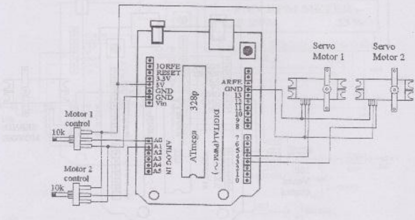
```
#include<Servo.h> //මෙමගින් Servo motor , Libraryය වෙත මෘදුකාංගය ගෙනයයි.

Servo myservo; // // ආරම්භයෙදී මෝටරය එහි ආරම්භක ස්ථානයේ පවතින බව දැනුම් දෙයි

int potpin = 0;  //  10k potentiometer ON ANALOG PIN 0
int val;   // variable to read the value from the analog pin

void setup()
{
 myservo.attach(8);  //  servo on pin 8
}

void loop()
{
 val = analogRead(potpin);          // reads the value of the potentiometer (value between
0 and 1023)
 val = map(val, 0, 1023, 0, 180);   // scale it to use it with the servo (value between 0
and 180)
 myservo.write(val);                // sets the servo position according to the scaled value
 delay(15);                         // waits for the servo to get there
}
```

// 10k පරිමා පාලකය දෙපසට කැරකැවීමෙන් මෝටරය අංශක 0 සිට අංශක 180 දක්වා දෙපසට භ්‍රමණය කරවීමට හැකිය.

මෙම නිර්මානයට අදාල sketch ය CD තැටියේ PROJECTS ලෙස දක්වා ඇති Folder යේ SERVO PSN තුල ලබාදී ඇත.

Project 19
පරීක්ෂණ අංක 19

Ardiuno පරිපථයක් භාවිතයෙන් SERVO MOTOR 2 ක් 10k පරිමා පාලක 2 ක් ආධාරයෙන් සිරස්ව හා තිරස්ව භ්‍රමනය කලහැකි සරල රොබෝ Arm කොටසක් සකසා ගන්නා ආකාරය සොයා බලමු. මේ සඳහා පහත දැක්වෙන පරිපථය සකසන්න.



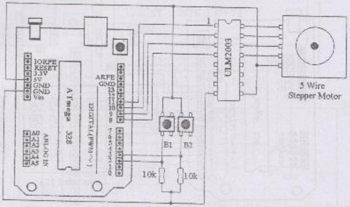
පහත සඳහන් ප්‍රෝග්‍රෑම කොටසින් SERVO MOTOR යුගලය සිරස්ව හා තිරස්ව භ්‍රමනය කිරීමට යොදාගතහැකිය.

```
void setup()
{
  HorizontalServo.attach(ServoHorizontalPin);
  VerticalServo.attach(ServoVerticalPin);
}

void loop()
{
  HorizontalPotValue = analogRead(HorizontalPotPin);
  VerticalPotValue = analogRead(VerticalPotPin);
  HorizontalServoPosition = map(HorizontalPotValue, 0, 1023, ServoH_Min ,
ServoH_Max);
  VerticalServoPosition = map(VerticalPotValue, 0, 1023, ServoH_Min ,
ServoH_Max);
  HorizontalServo.write(HorizontalServoPosition);
  VerticalServo.write(VerticalServoPosition);
  delay(20);
}
```

මෙම නිර්මාණයට අදාල sketch ඔ CD තැටියේ PROJECTS ලෙස දක්වා ඇති Folder යේ 2 SERVO ARM තුළ ලබාදී ඇත.

Project 20
පරීක්ෂණ අංක 20

Ardiuno පරිපථයක් භාවිතයෙන් වයර 5 ක් ඇති Stepper motor(Uni polar motor) යක් සුවිච 2 ක් (B1 , B2) ආධාරයෙන් දෙපසට භ්‍රමණය කරන ආකාරය සොයා බලමු.
පහත දැක්වෙන පරිපථය සකසන්න.



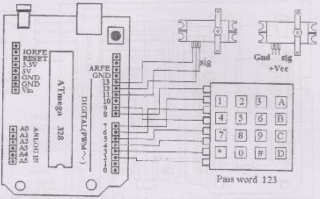
B1 ඔබනය එබීමෙන් පහත දැක්වෙන ආකාරයට Ardiuno පරිපථයේ 8,9,10,11 පින් අනු වලින් මෝටරය වාමාවර්ථ භ්‍රමනය වීමට ප්‍රෝග්‍රෑමය පහත දැක්වෙන ආකාරයට ලිවිය යුතුය.

```
val1 = digitalRead(button_1); // පළමුව B1 ඔබනය එබී ඇතිදැයි පරීක්ෂා කරයි.
if (val1 == HIGH)
{
digitalWrite(motorPin1, HIGH); // මෝටරය කැරකැවීමට  අදාල වෝල්ටියතා
ලබාදෙයි
digitalWrite(motorPin2, LOW);
digitalWrite(motorPin3, LOW);
digitalWrite(motorPin4, LOW);
delay(motor_Speed);
digitalWrite(motorPin1, LOW);
digitalWrite(motorPin2, HIGH);
digitalWrite(motorPin3, LOW);
digitalWrite(motorPin4, LOW);
delay(motor_Speed);
digitalWrite(motorPin1, LOW);
digitalWrite(motorPin2, LOW);
digitalWrite(motorPin3, HIGH);
digitalWrite(motorPin4, LOW);
delay(motor_Speed);
digitalWrite(motorPin1, LOW);
digitalWrite(motorPin2, LOW);
digitalWrite(motorPin3, LOW);
digitalWrite(motorPin4, HIGH);
delay(motor_Speed);
```

මෙම නිර්මානයට අදාල sketch ය CD තැටියේ PROJECTS ලෙස දක්වා ඇති Folder යේ Uni Polar Stepper Motor තුල ලබාදී ඇත.

Project 21
පරීක්ෂණ අංක 21

මෙම පරීක්ෂණයේදී Ardiuno Uno පරිපථයකට ඈඳුම්කල KEY PAD යක් ආධාරයෙන් Servo Motor යක් අංශක 0^0 සිට අංශක 180^0 ක් භ්‍රමනය කිරීමටත් එය නැවත අංශක 0^0 වෙත ගෙන ඒමටත් හැකි පරිපථයක් හා ප්‍රෝග්‍රැමයක් සකසන ආකාරය සොයා බලමු.



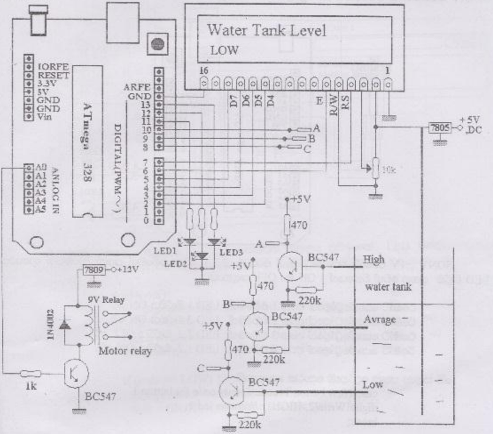
පහත සඳහන් ප්‍රෝග්‍රෑම් කොටසින් Key pad වෙත ලබාදෙන රහස් අංකයට (123) අනුව servo motor යේ ක්‍රියාකාරීත්වය පාලනය කරයි.

```
void loop()
{
char key = keypad.getKey();
if (key == '*'|| key == '#')
{                                   key pad යේ * හෝ # එබී ඇතිනම් මෝටරය
                                    අංශක 0 වෙත ගෙන එයි
position = 0;
setLocked(true);
}
if (key == password[position]) // රහස් අංකය ( 123 ) එබී ඇතිනම් මෝටරය
                                   භ්‍රමණය කිරීමට විධාන කරයි
{
position ++;}
if (position == 3){
setLocked(false);}
delay(100);}
void setLocked(int locked){
if (locked){
digitalWrite(redPin, HIGH);
digitalWrite(greenPin, LOW);
servo_Motor.write(11);}
else{
digitalWrite(redPin, LOW);
digitalWrite(greenPin, HIGH);
servo_Motor.write(180);
}
}
```

මෙම නිර්මාණයට අදාල sketch ය CD තැටියේ PROJECTS ලෙස දක්වා ඇති Folder යේ KEYPAD SERVO CONTROL තුල ලබාදී ඇත.

පරීක්ෂණ අංක 22

Ardiuno Uno පරිපථයක් මගින් ජල ටැංකියක ඇති ජල ප්‍රමාණය පෙන්වීමත් ජලය අඩුවීමකදී ජල පොම්පය ක්‍රියාත්මක කිරීමත් කලහැකි පරිපථයක් හා ප්‍රෝග්‍රෑමයක් සකසන ආකාරය සොයා බලමු.



පරිපථයේ A, B, C යන ස්ථාන එකිනෙකහා ඈඳුම් කරන්න. ටැංකියේ ඇති ජල ප්‍රමානය LCD පුවරුවේ පෙන්නුම් කරන අතර එය LED බල්බ 3 මගින්ද පෙන්නුම් කරයි.

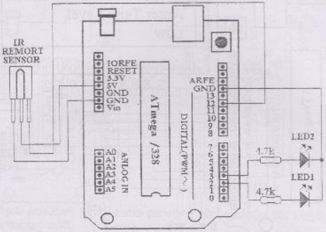
ජලය Low ස්ථානයට ආ විට LCD පුවරුවේ WATER LOW ලෙස පෙන්නුම් කරන අතර මේ සමගම ජල පෝම්පය ක්‍රියාත්මක කිරීම සඳහා Relay උපාංගය ON වේ.

ජලය High ස්ථානයට ආවිට LCD පුවරුවේ WATER HIGH ලෙස පෙන්නුම් කරන අතර මේ සමඟ පොම්පයේ ක්‍රියාකාරීත්වය නතර කිරීමට Relay උපාංගය OFF කරයි.

මෙම නිර්මානයට අදාල sketch ය CD තැටියේ PROJECTS ලෙස දක්වා ඇති Folder යේ WATER TANK LEVEL තුල ලබාදී ඇත.

Project 23
පරීක්ෂණ අංක 23

SONY , TV රිමෝට් කොන්ට්‍රෝලයක් භාවිතා කර LED බල්බ යුගලයක් වෙන් වශයෙන් ON හා OFF කරගතහැකි පරිපථයක් හා ප්‍රෝග්‍රෑමයක් සකසමු.



SONY , TV රිමෝට් කොන්ට්‍රෝලයේ අංක 1 ,2,3,4 බටන එබීමෙන් පහත දැක්වෙන ආකාරයට LED බල්බ යුගල වෙන් වශයෙන් ON හා OFF කලහැකිය.

රිමෝට් කොන්ට්‍රෝලයේ අංක 1 එබීමෙන් LED 1 බල්බය ON වේ.
රිමෝට් කොන්ට්‍රෝලයේ අංක 2 එබීමෙන් LED 2 බල්බය ON වේ.
රිමෝට් කොන්ට්‍රෝලයේ අංක 3 එබීමෙන් LED 1,2 බල්බය OFF වේ.
රිමෝට් කොන්ට්‍රෝලයේ අංක 4 එබීමෙන් LED 1,2 බල්බය ON වේ.

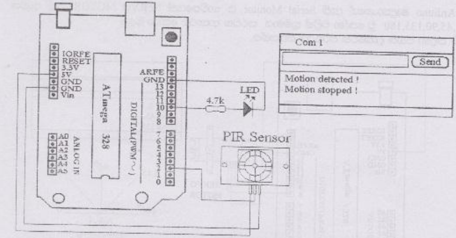
මේ සඳහා පහත ප්‍රෝග්‍රෑම කොටස යොදාගනී.

```
case 16: // button 1 code// sony code for button 1
digitalWrite(2, HIGH); //turn on led 1
break;

case 2064: //button 2 code// sony code for button 2
digitalWrite(3, HIGH); //turn on led 2
break;

case 1040: //button 3 code// sony code for button 3
digitalWrite(2, LOW); //turn off led 1
digitalWrite(3, LOW); //turn off led 2
break;

case 3088: //button 4 code// sony code for button 4
digitalWrite(2, HIGH); //turn off led 1
digitalWrite(3, HIGH); //turn off led 2
break;
```

මෙම නිර්මානයට අදාල sketch ය CD තැටියේ PROJECTS ලෙස දක්වා ඇති Folder යේ IR LED ON OFF තුල ලබාදී ඇත.

පරීක්ෂණ අංක 24

PIR සෙන්සර් උපාංගයක් භාවිතාකර යම් වස්තුවක චලනය වීමට අනුරූපීව LED බල්බයක් ON වීමටත් එය Ardiuno Uno මෘදුකාංගයේ ඇති Serial monitor යේ පෙන්වීමටත් හැකි පරිපථයක් හා ප්‍රෝග්‍රෑමයක් සකසමු.



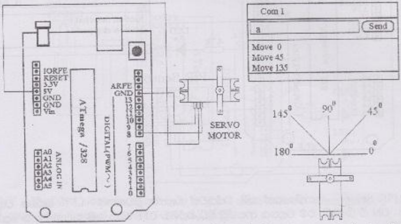
මෙම PIR Sensor ආසන්නයෙන් යම් වස්තුවක් චලනය වුවහොත් LED බල්බය චලනය අවසන් වනතුරු ON වී තිබෙයි. එම චලනය ආසන්වූ විට බල්බය OFF වෙයි. මේ සඳහා පහත ප්‍රෝග්‍රෑම් කොටස යොදාගනී.

```
void loop(){
 val = digitalRead(sensor);  // read sensor value
 if (val == HIGH) {          // check if the sensor is HIGH
  digitalWrite(led, HIGH);  // turn LED ON
  delay(100);               // delay 100 milliseconds

  if (state == LOW) {
   Serial.println("Motion detected!");
   state = HIGH;     // update variable state to HIGH
  }
 }
 else {
   digitalWrite(led, LOW); // turn LED OFF
   delay(200);          // delay 200 milliseconds

   if (state == HIGH){
    Serial.println("Motion stopped!");
    state = LOW;      // update variable state to LOW
  }
 }
}
```

මෙම නිර්මානයට අදාල sketch ය CD තැටියේ PROJECTS ලෙස දක්වා ඇති Folder යේ PIR SENSOR ALARM තුල ලබාදී ඇත.

Project 25
පරීක්ෂණ අංක 25

Arduino මෘදුකාංගයේ ඇති Serial Monitor ය භාවිතයෙන් SERVO MOTOR යක් අංශක 0 ,45,90,135,180 වූ කෝණ වලින් භ්‍රමණය කරවන ආකාරය සොයා බලමු.
මේ සඳහා පහත දැක්වෙන පරිපථය සකසන්න.

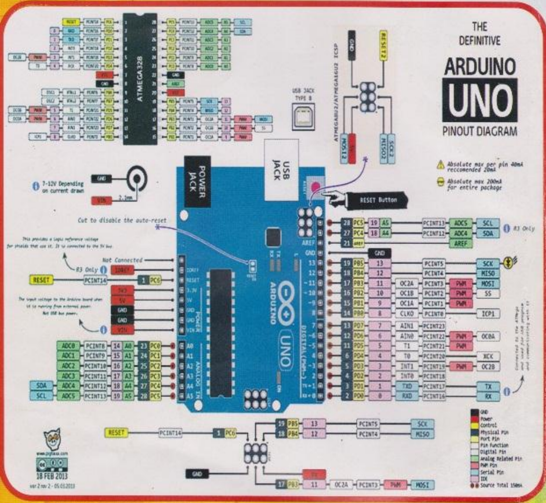


පරිපථයට විභවය ලබාදුන්විට මෝටරය එහි ආරම්භක ස්ථානයේ වෙත (අංශක 0 වෙත) ගමන් කරයි.

ඇත් Serial Monitor යට a,s,d,f,g අක්ෂර key board ය භාවිතයෙන් ලබාදුන්විට ඊට අනුරූපී කෝණය වෙත මෝටරය භ්‍රමනය වී නතරවෙයි. භ්‍රමණය වන කෝණය Serial Monitor යේ පෙන්නුම් කරයි.

```
void loop(){
 input = Serial.read();
  if (input == 'a' ) // Serial Monitor යේ "a" අක්ෂරය සටහන් කලවිට මෝටරය අංශක 0 වෙත යයි {
   myservo.write(0);
   delay(25); }
   Serial.println("MOVE 0"); // ගමන් කළ කෝණය පෙන්නුම් කරයි
   }
   if (input == 's') // Serial Monitor යේ "s" අක්ෂරය සටහන් කලවිට මෝටරය අංශක 45 වෙත යයි
    {
    myservo.write(45);
    delay(25); }
     Serial.println("MOVE 45"); //ගමන් කළ කෝණය පෙන්නුම් කරයි
```

මෙම නිර්මානයට අදාල sketch ය CD තැටියේ PROJECTS ලෙස දක්වා ඇති Folder යේ SERVO SE COUNT තුළ ලබාදී ඇත.



කර්තෘ ප්‍රකාශනයකි

මිල
රු. 490/-