Embedded Workshop
Feb 24, 2016

Rusty Cain

Set up for Workshop:

Please Sign in on Sheet. Please include your email.

While you are waiting for the Workshop to begin…

1. Make sure you are connected to the local Wifi
   Guest Password: **Welcome2DMS**
2. Make sure you have **Arduino IDE** installed and working:
   Download **Arduino Version 1.6.4** or newer.
   Add Libraries
   Sketch - Include Library - Manage Libraries
   <Keypad.h> <LiquidCrystal_I2C.h> <Wire.h>
3. **Copy Programs off the USB Memory stick.**
   Project1 - Project2 - Project3 etc…

Parts Needed: (Ask about Parts kits)

- 1 - Arduino Uno, Breadboard & Wires,
- 1 - 3X4 Keypad
- 1 - I2C bus LCD Display, 20x4 or 16x2
- 2 - 330 ohm resistor
- 1 - LED Red
- 1 - LED Green

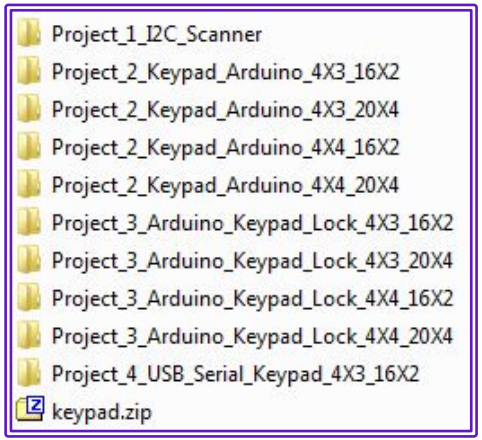
http://manitou-solutions.com/ewm/

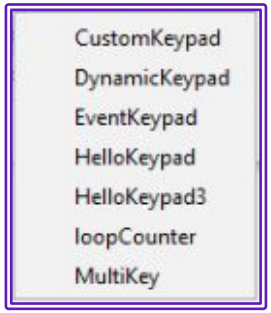
Please sign in

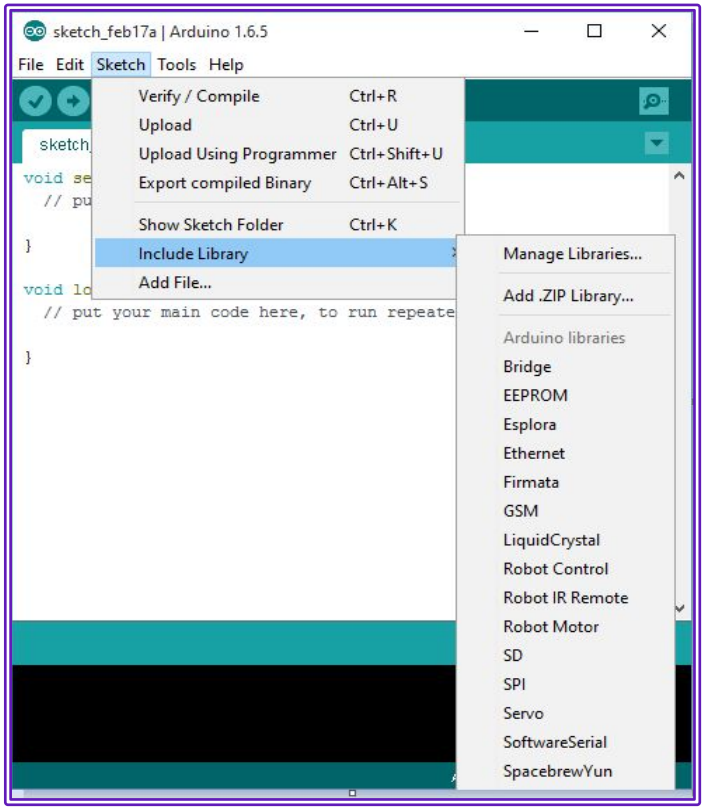
Name	Member Y/N	Email address

The Keypad Library

http://playground.arduino.cc/uploads/Code/keypad.zip







Download here: keypad.zip

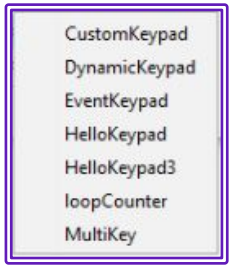
Put the Keypad folder in "*arduino*\libraries\".

In the Arduino IDE, create a new sketch (or open one) and select from the menubar "Sketch -> Import Library -> Keypad".

Once the library is imported, an "#include <Keypad.h>" line will appear at the top of your Sketch.

Keypad Library functions

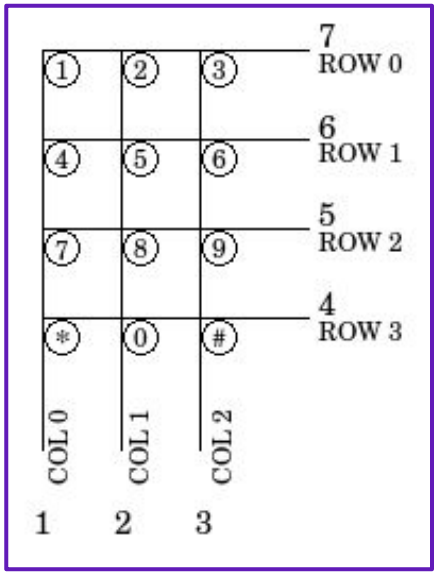
```
Functions
void begin(makeKeymap(userKeymap))
Initializes the internal keymap to be equal to userKeymap
[See File -> Examples -> Keypad -> Examples -> CustomKeypad]

char waitForKey()
This function will wait forever until someone presses a key.
Warning: It blocks all other code until a key is pressed.
That means no blinking LEDs, no LCD screen updates, no nothing with the exception of interrupt routines.

char getKey()
Returns the key that is pressed, if any. This function is non-blocking.

KeyState getState()
Returns the current state of any of the keys.
The four states are IDLE, PRESSED, RELEASED and HOLD.

boolean keyStateChanged()
New in version 2.0: Let's you know when the key has changed from one state to another.
For example, instead of just testing for a valid key you can test for when a key was pressed.

setHoldTime(unsigned int time)
Set the amount of milliseconds the user will have to hold a button until the HOLD state is triggered.

setDebounceTime(unsigned int time)
Set the amount of milliseconds the keypad will wait until it accepts a new keypress/keyEvent.
This is the "time delay" debounce method.

addEventListener(keypadEvent)
Trigger an event if the keypad is used. You can load an example in the Arduino IDE.
[See File -> Examples -> Keypad -> Examples -> EventSerialKeypad] or see the KeypadEvent Example code.
```



Reading a Keypad with Arduino

Keypads consist of **Normally Open** switches that connect a row with a column when pressed.
Each of the four rows are connected to an **input** pin.
Each of the (3 or 4) columns are connected to an **output** pin.

The **getKey** function sequentially sets the pin for each column **LOW** and then reads to see if any of the row pins are **LOW.**
Closing a switch produces a **LOW** signal on the input pin.
If they are **LOW** this indicates that the switch for that row and column is closed.

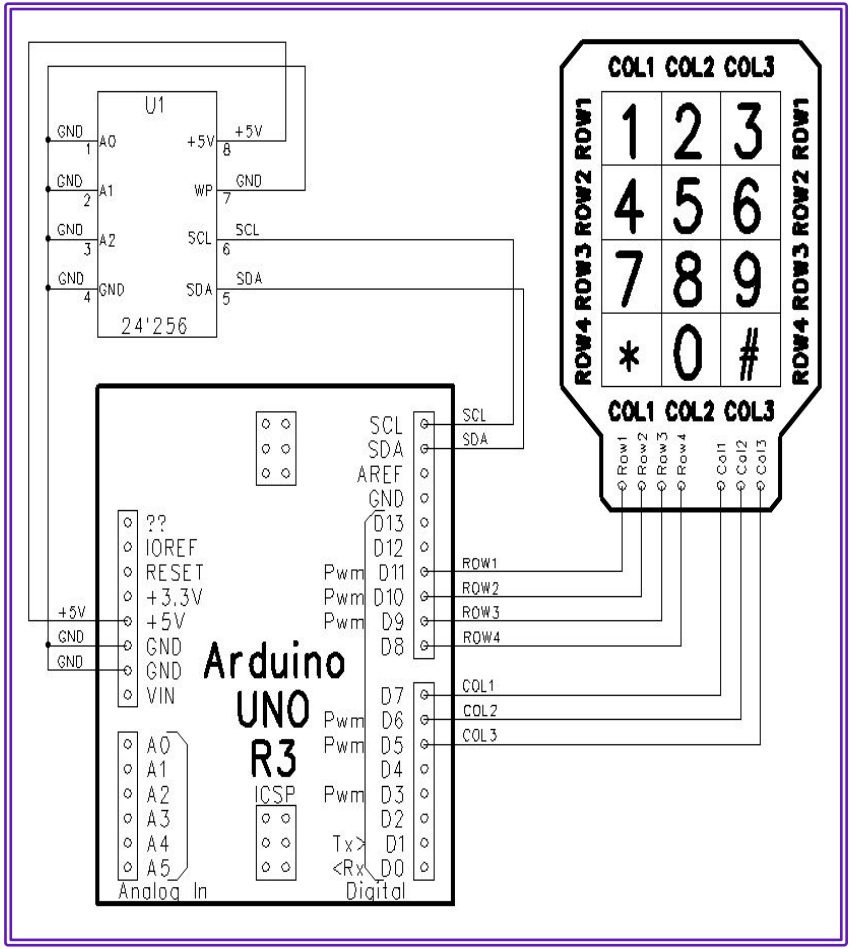


Connecting KeyPad to Arduino

Connect to Arduino as shown in the diagram

```
const byte rows = 4;
const byte cols = 4;

char keys[rows][cols] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};

byte rowPins[rows] = {11,10,9,8};
byte colPins[cols] = {7,6,5,4};
```

3X4 Keypad	Arduino Pin	Row Column	4X4 KeyPad	Arduino Pin	Row Column
			8	4	Col 4
7	5	Col 3	7	5	Col 3
6	6	Col 2	6	6	Col 2
5	7	Col 1	5	7	Col 1
4	8	Row 4	4	8	Row 4
3	9	Row 3	3	9	Row 3
2	10	Row 2	2	10	Row 2
1	11	Row 1	1	11	Row 1

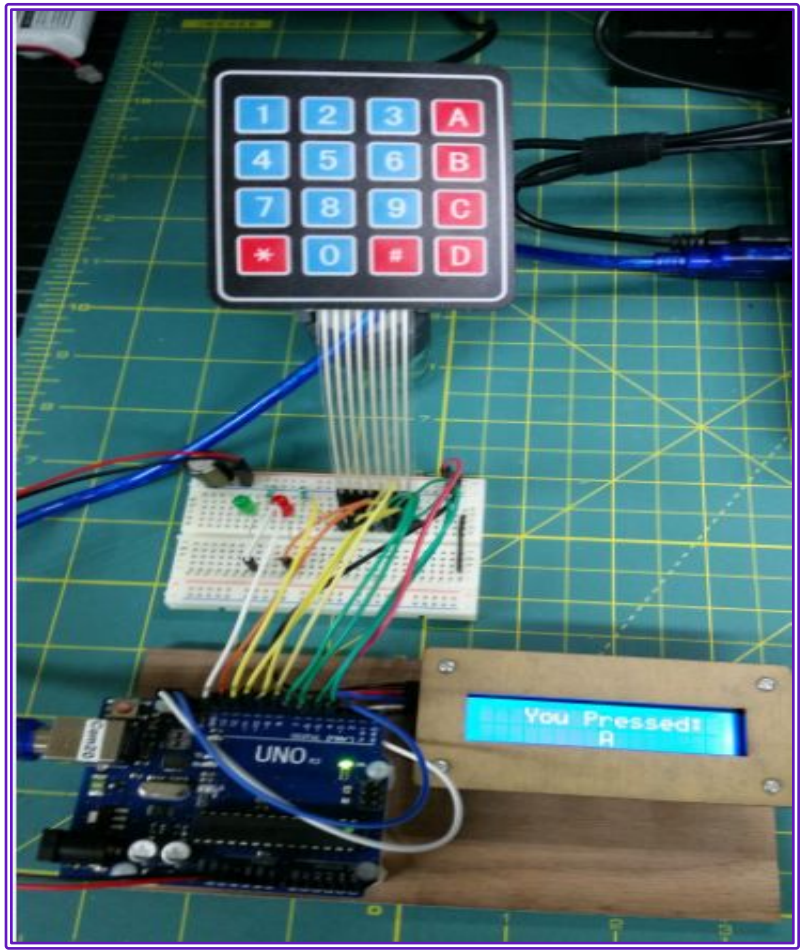


Project 2 Let's get Started!

Download Library
Connect Arduino to Keypad
Connect I2C LCD (I2C Scanner) or Use Serial Monitor
Load one of the following Sketch's

- Project_2_Keypad_Arduino_4X3_16X2
- Project_2_Keypad_Arduino_4X3_20X4
- Project_2_Keypad_Arduino_4X4_16X2
- Project_2_Keypad_Arduino_4X4_20X4

Press any key on the Keypad:



```
##############################################################################*/

#include <Wire.h>                       // Libarary for the I2C LCD
#include <LiquidCrystal_I2C.h>          // Libarary for the I2C LCD
#include <Keypad.h>                     // Libarary for the Keypad

const byte rows = 4;                    // Four rows
const byte cols = 3;                    // Select for Three columns
//const byte cols = 4;                  // Select for Four columns

/*char keys[rows][cols] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
}; */

char keys[rows][cols] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}
};
byte rowPins[rows] = {11,10,9,8};       // Connect to the row pinouts of the keypad
byte colPins[cols] = {7,6,5};           // For 3 column - connect to the column pinouts of the keypad
//byte colPins[cols] = {7,6,5,4};       // For 4 column - connect to the column pinouts of the keypad

Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, rows, cols);

LiquidCrystal_I2C lcd(0x27,20,4);       // Set the LCD address to 0x27 for a 20 chars and 4 line display
//LiquidCrystal_I2C lcd(0x27,16,2);     // Set the LCD address to 0x27 for a 16 chars and 2 line display
```

```
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, rows, cols); //initialize an instance of class Keypad

LiquidCrystal_I2C lcd(0x27,20,4);       // Set the LCD address to 0x27 for a 20 chars and 4 line display
//LiquidCrystal_I2C lcd(0x27,16,2);     // Set the LCD address to 0x27 for a 16 chars and 2 line display

void setup()
{
  keypad.setDebounceTime(100);          // setDebounceTime(mS)
  lcd.init();                           // Initialize the lcd
  lcd.backlight();                      // Turn on Display backlight
  lcd.setCursor(3,0);                   // Place cursor Row 0 Column 3
  lcd.print("You Pressed:");            // Print Message
}

void loop()
{
  char key = keypad.getKey();           // Get Key
  if (int(key) != 0) {                  // Was any key pressed?
    lcd.setCursor(16,0);                // Set Cursor to Row 0 Column 16
    lcd.print(key);                     // Print key that was pressed
  }
}
/*##########################################################################################################
                Serial Monitor instead of I2C LCD
##########################################################################################################
void setup(){
  Serial.begin(9600);                   // init print
}
void loop()
{
  char key = keypad.getKey();           // Get Key
  if (int(key) != 0) {                  // Was any key pressed?
    Serial.println(key);                // Print key that was pressed
  }

##########################################################################################################*/
```

Project 3 - Keypad Lock

Connect LED's to Arduino

PIN 12 to RED LED & 330 OHM Resistor

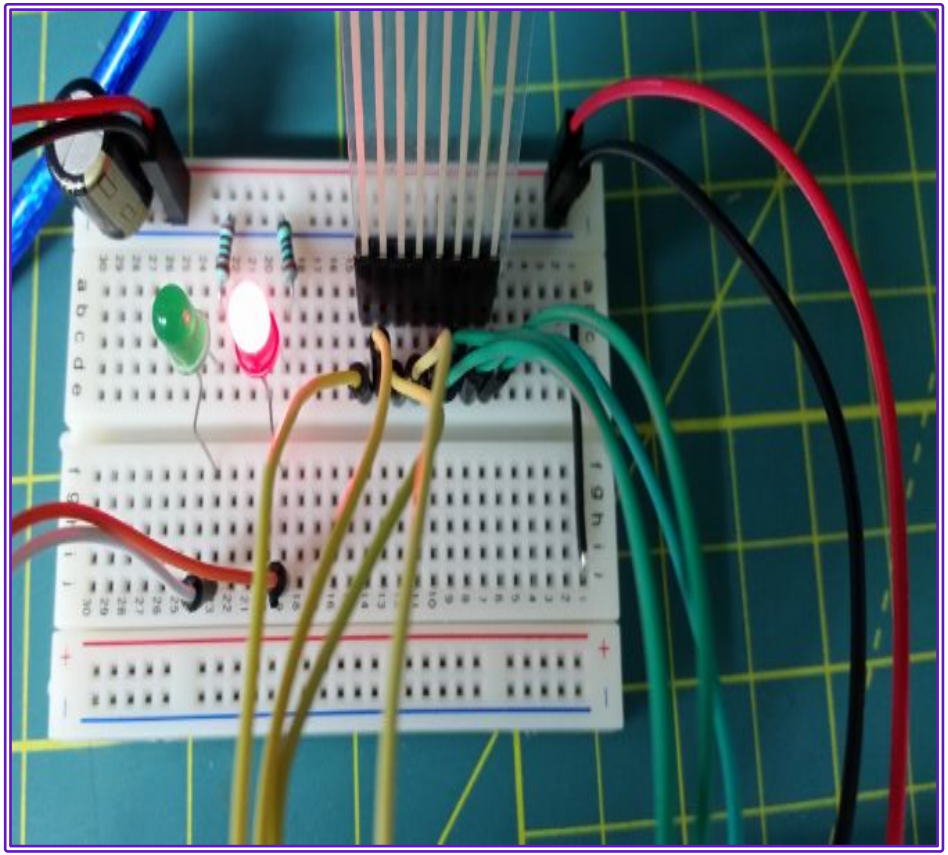
PIN 13 to GREEN LED & 330 OHM Resistor

Load one of the following Sketch's

- Project_3_Arduino_Keypad_Lock_4X3_16X2
- Project_3_Arduino_Keypad_Lock_4X3_20X4
- Project_3_Arduino_Keypad_Lock_4X4_16X2
- Project_3_Arduino_Keypad_Lock_4X4_20X4

Password: 2016

Let's Examine the code



Keypad lock

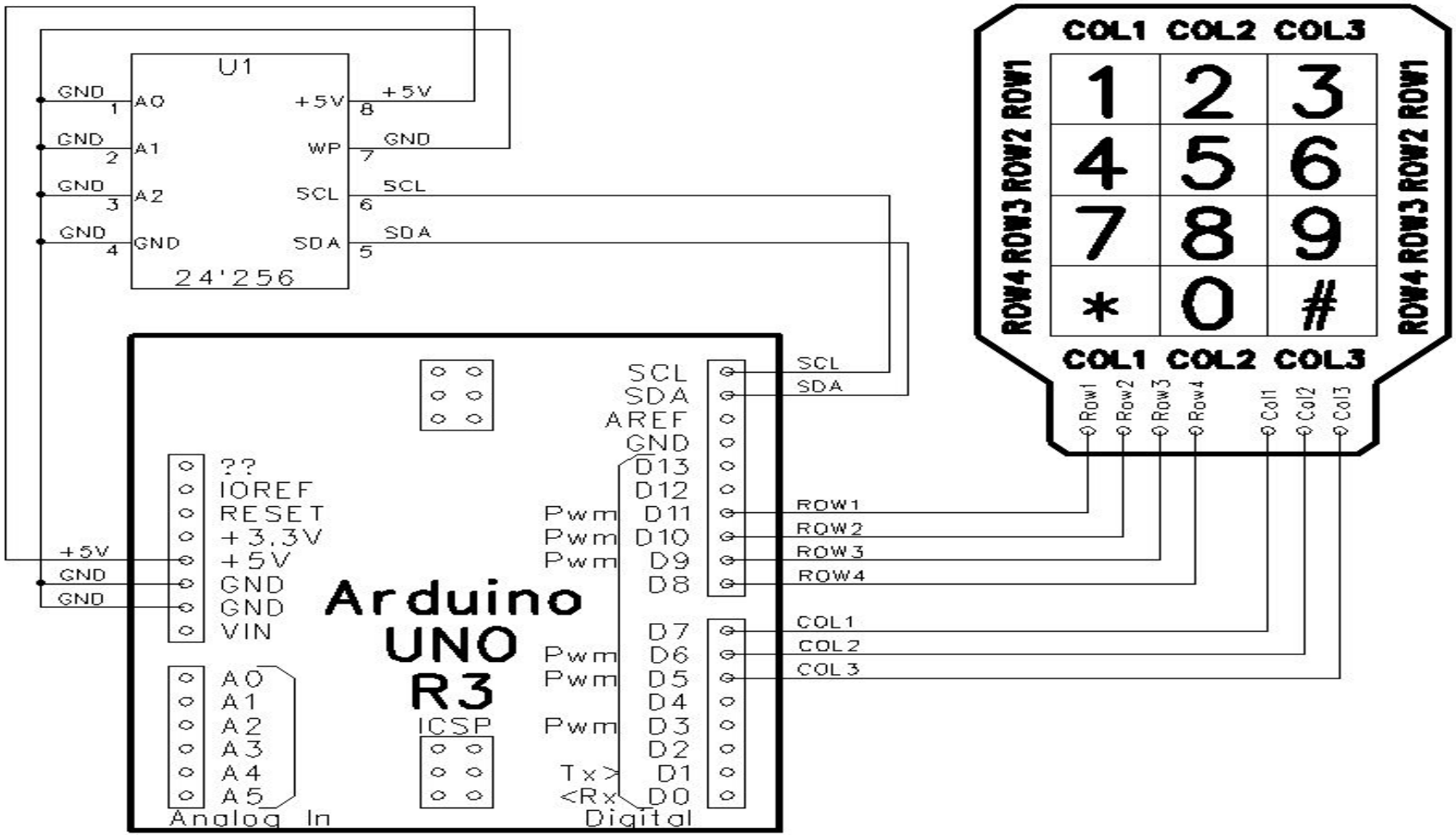
```
#include <Wire.h>                                  // Libarary for the I2C LCD
#include <LiquidCrystal_I2C.h>                     // Libarary for the I2C LCD
#include <Keypad.h>                                // Libarary for the Keypad

char* password = "2016";                           // Change the password here, just pick any 3 numbers
int curpost = 0;                                   // Track current position of keystroke
int Red_Lock = 12;                                 // Keylock Status = LOCKED
int Green_Unlock = 13;                             // Keylock Status = UN-LOCKED
const byte rows = 4;                               // Define Rows of keypad
const byte cols = 4;                               // Define Columns of keypad
//const byte cols = 3;                             // Define Columns of keypad
//  4X4 Keypad
char keys[rows][cols] = {
{'1','2','3','A'},
{'4','5','6','B'},
{'7','8','9','C'},
{'*','0','#','D'}
};

/* 4X3 Keypad
char keys[rows][cols] = {
{'1','2','3'},
{'4','5','6'},
{'7','8','9'},
{'*','0','#'}
};*/
 byte rowPins[rows] = {11,10,9,8};                 // Connect to the row pinouts of the keypad
// byte colPins[cols] = {7,6,5};                   //For 3 column - connect to the column pinouts of the keypad
 byte colPins[cols] = {7,6,5,4};                   // For 4 column - connect to the column pinouts of the keypad

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, rows, cols ); // Initialize an instance of class Keypad

LiquidCrystal_I2C lcd(0x27,20,4);                 // Set the LCD address to 0x27 for a 20 chars and 4 line display
// LiquidCrystal_I2C lcd(0x27,16,2);                 // Set the LCD address to 0x27 for a 16 chars and 2 line display
```

Keypad lock

```
void setup()
{
  pinMode(Red_Lock, OUTPUT);                      // Set up Red LED output
  pinMode(Green_Unlock, OUTPUT);                  // Set up Green LED output
  Serial.begin (9600);                            // init Serial Console
  Serial.println ();                              // Print lineFeed
  Serial.println (" Project_3_Arduino_Lock_4X4 "); // Print to Serial Console
  Serial.println (" Embedded Workshop ");         // Print to Serial Console
  Serial.println (" Feb 24, 2016 -  Rusty Cain"); // Print to Serial Console
  Serial.println (" Display - LCD 16X2 ");        // Print to Serial Console
  Serial.println (" Keyboard 4X4 ");              // Print to Serial Console
  lcd.init();                                     // Initialize the lcd
  displayEntry();                                 // Function call to Print a message to the LCD.
  lcd.backlight();                                // Turn on Display backlight
  }

void loop()
{
  digitalWrite(Red_Lock, HIGH);                   // Turn On Red LED to show LOCKED status
  int a;                                          // Loop Counter
  char key = keypad.getKey();

 if (int(key) != 0) {                             // Get Key
      lcd.setCursor(6,1);                         // Place Cursor on LCD
      lcd.print("#");                             // LCD Print
      delay (500);                                // Delay
      lcd.setCursor(6,1);                         //Place Cursor
      lcd.print(" ");                             // LCD Print
    for (a=0; a == curpost; ++a)                  // Loop for comparing data entered
      {
          //      lcd.setCursor(a,2);
          //      lcd.print("?");
      }
   if (key == password[curpost])                  // Compare Password
      {
         ++curpost;                               // add to the count
      if (curpost == 4)                           // Compare if last digit
         {
          unlock();                               // Passcode is correct go to unlock function
          curpost = 0;                            // Reset count
         }
      } else {
           invalidcode();                         // Wrong passcode go to invalidcode function
           curpost = 0;                           // Reset count
            }
  }
}
```

```
void displayEntry()
{
  clearScreen();
  lcd.setCursor(0,0);
  lcd.print("    Enter Code:     ");
  lcd.setCursor(0,1);
  lcd.print("     ");
  lcd.setCursor(0,2);
  lcd.print("**  DMS Workshop  **");
  lcd.setCursor(0,3);
  lcd.print("                    ");
}
void clearScreen()
{
  lcd.setCursor(0,0);
  lcd.print("                    ");
  lcd.setCursor(0,1);
  lcd.print("                    ");
  lcd.setCursor(0,2);
  lcd.print("                    ");
  lcd.setCursor(0,3);
  lcd.print("                    ");
}
```

```
void invalidcode()
{
 digitalWrite(Red_Lock, HIGH);
  clearScreen();
  lcd.setCursor(0,0);
  lcd.print(" Access Denied!!!!  ");
  lcd.setCursor(0,1);
  lcd.print("   Red LED On       ");
  delay(1500);
  lcd.setCursor(0,2);
  lcd.print("**  DMS Workshop  **");
  lcd.setCursor(0,3);
  lcd.print("                    ");
  delay (1000);
  digitalWrite(Red_Lock, LOW);
  displayEntry();
}

void unlock()
  {
  digitalWrite(Red_Lock, LOW);
  digitalWrite(Green_Unlock, HIGH);
  clearScreen();
  lcd.setCursor(0,0);
  lcd.print(" Access Granted!!!! ");
  lcd.setCursor(0,1);
  lcd.print(" Green LED on       ");
  delay(1000);
  lcd.setCursor(0,2);
  lcd.print("**  DMS Workshop  **");
  lcd.setCursor(0,3);
  lcd.print("                    ");
  delay (5000);
  digitalWrite(Green_Unlock, LOW);
  displayEntry();
  }
```


U1
GND
A0
1
GND
A1
2
GND
A2
3
GND
GND
4
+5V
+5V
8
WP
GND
7
SCL
SCL
6
SDA
SDA
5
24'256
COL1 COL2 COL3
ROW1
ROW2
ROW3
ROW4
1 2 3
4 5 6
7 8 9
* 0 #
Row1 Row2 Row3 Row4 Col1 Col2 Col3
SCL
SDA
AREF
GND
D13
D12
Pwm D11
Pwm D10
Pwm D9
D8
D7
Pwm D6
Pwm D5
D4
Pwm D3
D2
Tx> D1
<Rx D0
Digital
ROW1
ROW2
ROW3
ROW4
COL1
COL2
COL3
??
IOREF
RESET
+3.3V
+5V
GND
GND
VIN
+5V
GND
GND
A0
A1
A2
A3
A4
A5
Analog In
Arduino
UNO
R3
ICSP

Project 6 I2C/TWI 1602 Serial LCD Module Display

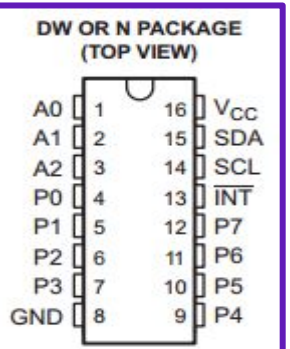
PCF8574AN I2C Address:

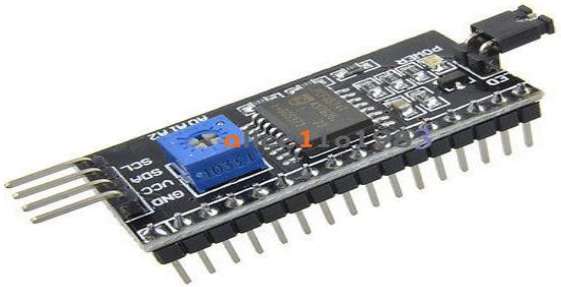
Board 0x20~0x27

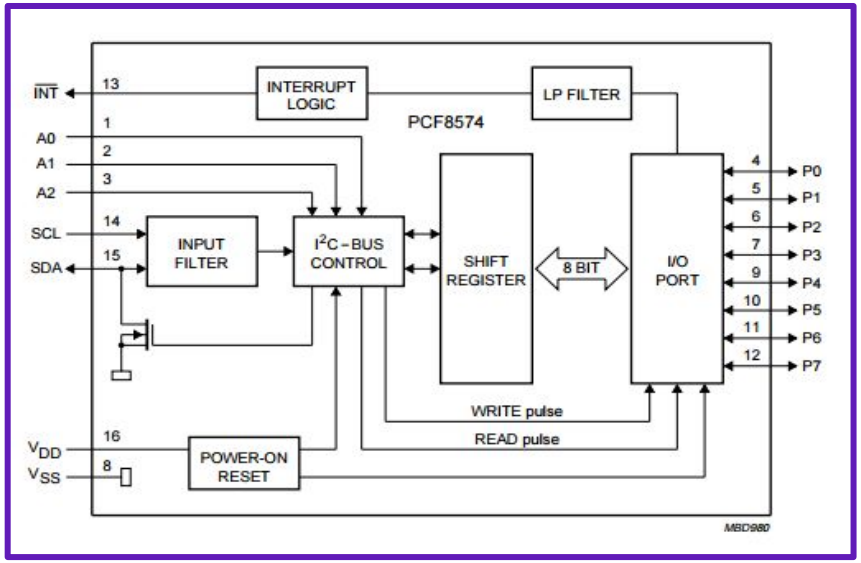
Chip 0x38 - 0x3F

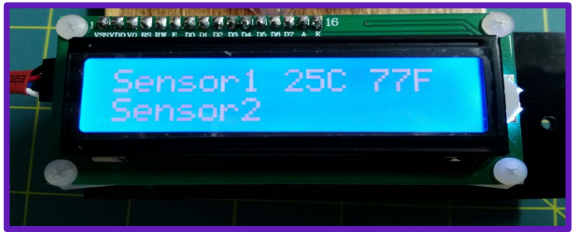
Backlight and contrast is adjusted by potentiometer

```
#include  <Wire.h>
#include <LiquidCrystal_I2C.h>
```









Notes: