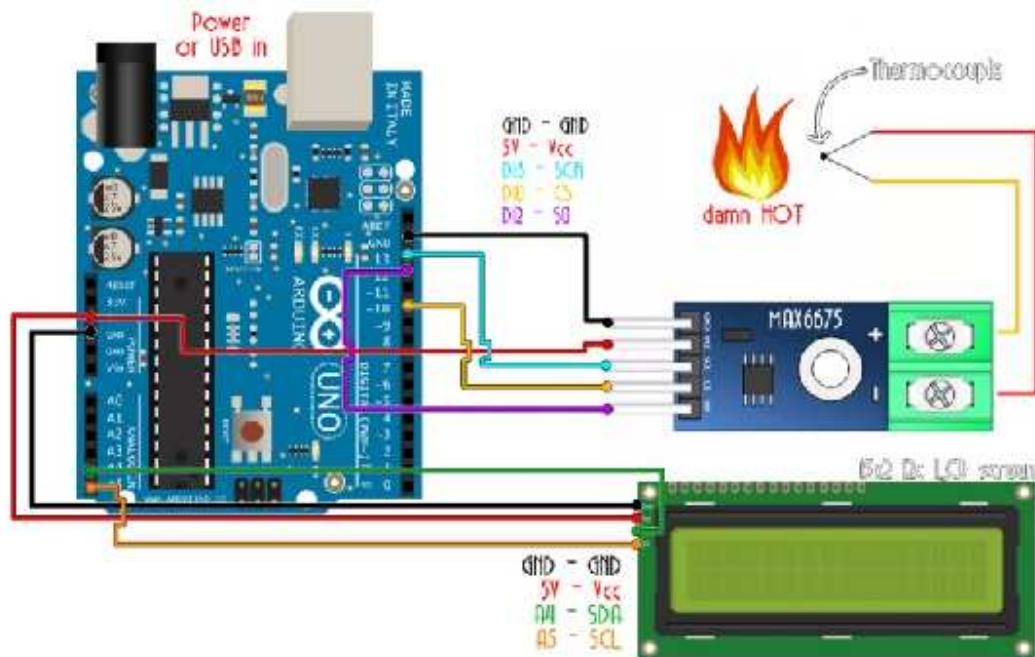


ARDUINO TEMPERATURE INDICATOR WITH MAX6675 & I2C-LCD-16x2 DISPLAY



Program

TESTED OK BY MR. ADEEB RAZA

- * PID temperature controller DIY Arduino
- * http://www.electrooobs.com/eng_ardui...
- *
- * Max6675 Module ==> Arduino
- * CS ==> D10
- * SO ==> D12
- * SCK ==> D13
- * Vcc ==> Vcc (5v)
- * Gnd ==> Gnd */

//LCD config

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

```
#include <LiquidCrystal_I2C.h> //If you don't have the LiquidCrystal_I2C library, download it and install it
```

```
LiquidCrystal_I2C lcd(0x27,16,2); //sometimes the adress is not 0x3f. Change to 0x27 if it doesn't work.
```

```
/* i2c LCD Module ==> Arduino
```

```
* SCL ==> A5
```

```

* SDA      ==>  A4
* Vcc      ==>  Vcc (5v)
* Gnd      ==>  Gnd  */

#include <SPI.h>

#define MAX6675_CS  10
#define MAX6675_SO  12
#define MAX6675_SCK 13

void setup() {
  lcd.init();
  lcd.backlight();
}

void loop() {
  float temperature_read = readThermocouple();
  lcd.setCursor(0,0);
  lcd.print("TEMPERATURE");
  lcd.setCursor(7,1);
  lcd.print(temperature_read,1);
  delay(1000);
}

double readThermocouple() {

  uint16_t v;
  pinMode(MAX6675_CS, OUTPUT);
  pinMode(MAX6675_SO, INPUT);
  pinMode(MAX6675_SCK, OUTPUT);

  digitalWrite(MAX6675_CS, LOW);
  delay(1);

  // Read in 16 bits,
  // 15 = 0 always
  // 14..2 = 0.25 degree counts MSB First
  // 2 = 1 if thermocouple is open circuit
  // 1..0 = uninteresting status

  v = shiftIn(MAX6675_SO, MAX6675_SCK, MSBFIRST);
  v <<= 8;
  v |= shiftIn(MAX6675_SO, MAX6675_SCK, MSBFIRST);

  digitalWrite(MAX6675_CS, HIGH);
  if (v & 0x4)
  {

```

```

// Bit 2 indicates if the thermocouple is disconnected
return NAN;
}

// The lower three bits (0,1,2) are discarded status bits
v >>= 3;

// The remaining bits are the number of 0.25 degree (C) counts
return v*0.25;
}

```

The End

