

```

/*
  arduino dual channel reader
*/
#include <LCD_I2C.h>
#include <Wire.h>

// These constants won't change. They're used to give names to the pins used:
const int analogInPin0 = A0; // Analog input pin that the potentiometer is
attached to
const int analogInPin1 = A1; // Analog input pin that the potentiometer is
attached to
const int analogOutPin = 9; // Analog output pin that the LED is attached to
int sensorValue0 = 0; // value read from the pot
int sensorValue1 = 0; // value read from the pot
int outputValue0 = 0; // value output to the PWM (analog out)
int outputValue1 = 0; // value output to the PWM (analog out)
int ch0_lrv;
int ch1_lrv;
int ch0_urv;
int ch1_urv;
LCD_I2C lcd(0x27, 16, 2); // Default address of most PCF8574 modules, change
according

void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
  lcd.begin(); // If you are using more I2C devices using the Wire library use
lcd.begin(false)
  lcd.backlight();
}

void loop(){
  //read the pushbutton value into a variable
  int sensorVal1 = digitalRead(2);
  int sensorVal2 = digitalRead(3);
  int sensorVal3 = digitalRead(4);
  int sensorVal4 = digitalRead(5);
  while ((sensorVal2) == LOW && (sensorVal3) == LOW)
  {
    lcd.clear();
    mainclockdisplay();
  }

  if (sensorVal1 == HIGH) {
    digitalWrite(13, LOW);
  } else {
    digitalWrite(13, HIGH);
    if(menu<10){
      menu=menu+1;
      Serial.println("MENU");
    }
  }
}

```

```

Serial.println(menu);
delay(1000);
}else{menu=1;}
}

switch (menu) {
case 1:
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("ch0 hihi setpoint"); // You can make spaces using well... spaces
lcd.setCursor(5, 1); // Or setting the cursor in the desired position.
lcd.print(RTC.getHourMode());
updn=RTC.getHourMode();
//dn=RTC.getHours();
if (sensorVal2 == HIGH) {
digitalWrite(13, LOW);
} else {
digitalWrite(13, HIGH);
if(updn<5){
updn=updn+1;
Serial.println("UP");
Serial.println(updn);
delay(1000);
}else{updn=0;}
}
if (sensorVal3 == HIGH) {
digitalWrite(13, LOW);
} else {
digitalWrite(13, HIGH);
if(updn<5){
updn=updn-1;
Serial.println("DN");
Serial.println(updn);
delay(1000);
}else{updn=0;}
}
lcd.setCursor(0, 0);
lcd.print("ch0 hihi setpoint"); // You can make spaces using well... spaces
lcd.setCursor(5, 1); // Or setting the cursor in the desired position.
RTC.setHourMode(updn);
lcd.print(RTC.getHourMode());
// updn=RTC.getHours();
delay(500);
break;
}

void loop1() {
// read the analog in value:
sensorValue0 = analogRead(analogInPin0);

```

```
// map it to the range of the analog out:
outputValue0 = map(sensorValue0, 0, 1023, ch0_lrv, ch0_urv);
// change the analog out value:
//analogWrite(analogOutPin, outputValue);
  sensorValue1 = analogRead(analogInPin1);
// map it to the range of the analog out:
outputValue1 = map(sensorValue1, 0, 1023, ch1_lrv, ch1_urv);
delay(1000);
// print the results to the Serial Monitor:
Serial.print("sensor0 = ");
Serial.print(sensorValue0);
Serial.print("\t output = ");
Serial.println(outputValue0);
Serial.print("sensor1 = ");
Serial.print(sensorValue1);
Serial.print("\t output = ");
Serial.println(outputValue1);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("1:");
lcd.setCursor(2, 0);
lcd.print(outputValue0); // You can make spaces using well... spaces
lcd.setCursor(3, 1); // Or setting the cursor in the desired position.
lcd.print(outputValue1);
//delay(1000);

// wait 2 milliseconds before the next loop for the analog-to-digital
// converter to settle after the last reading:
```

```
}
```