2.6 Equipment testing

2.6.1 Basic testing in Arduino IDE

1. Connect the Arduino Uno to PC with proper USB cable.

   [Arduino Uno]

2. Open Arduino IDE program and open program with:

   Files – Examples – 01. Basics – Blink.ino

3. Make shure that you will set the proper settings (see fig. 1). From the menu choose:

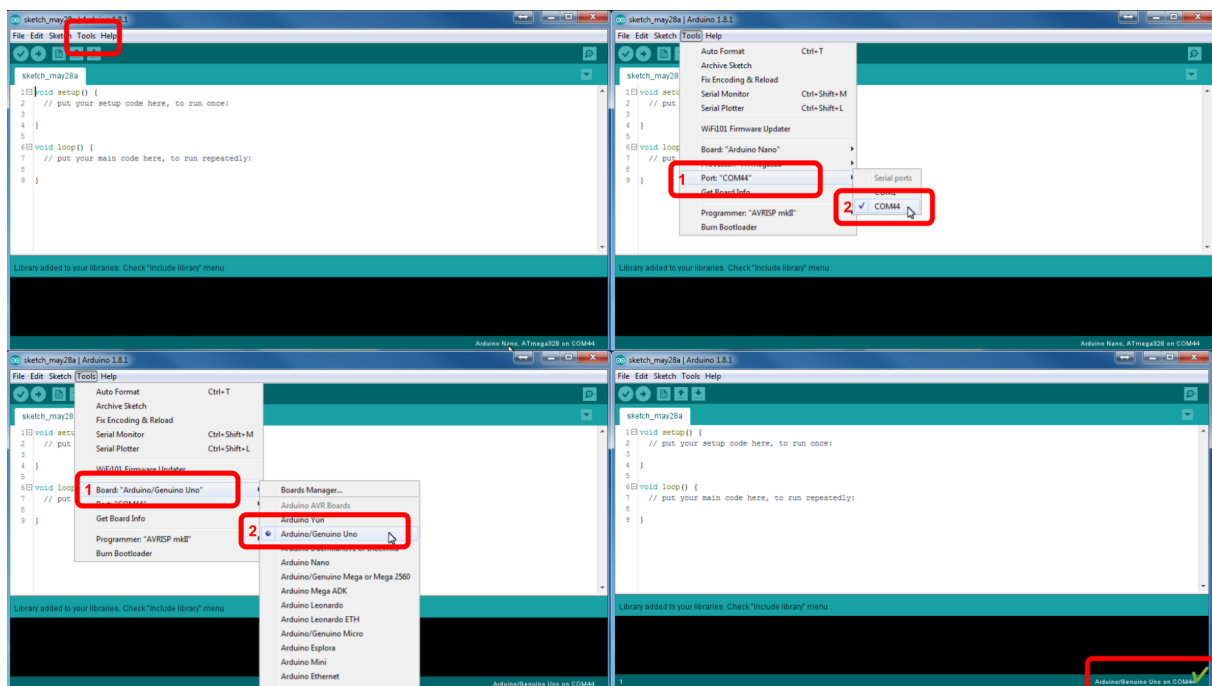   Tools-

   1. Board: Arduino/Genuino Uno
   2. Port: COM3

Figure 1: Arduino basic setup.

- To upload the code you can click the icon Upload.

  If the uploading was successful you will be prompted with the text like:

```
Done uploading.
Sketch uses 970 bytes (3%) of program storage space. Maximum
is 32256 bytes. Global variables use 9 bytes (0%) of dynamic
memory, leaving 2039 bytes for local variables. Maximum is
2048 bytes.
```
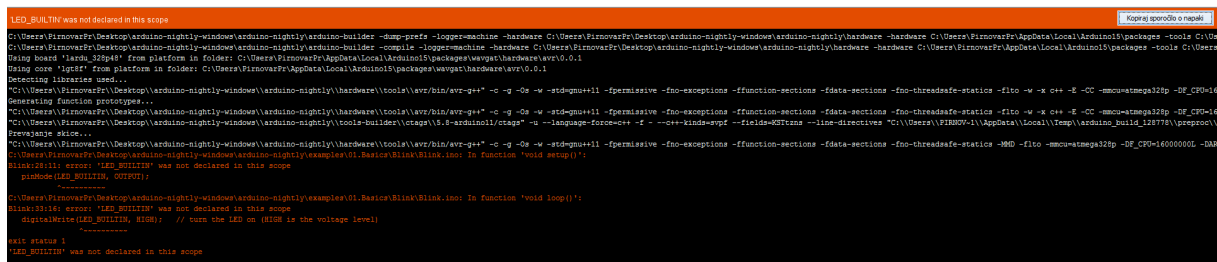
2.6.2 Issues

Figure 2: Error image.

2.6.2.1 LED_BUILTIN was not declared in this scope Compiler ne ve kaj naj bi bilo "LED_BUILTIN" … na tem mesu naj bi bila številka priključka, ki ga želimo krmiliti. V tem primeru je to številka 13. Rešitvi sta lahko 2:

1. vse LED_BUILTIN zamenjaš s 13 ali

2. v vrstico pred "void setup()" dodaj **const int** LED_BUILTIN = 13;

Zadnja (druga) rešitev je boljša, ker pripomore k berljivosti programa… Spremenljivka LED_BUILTIN se imenuje "razlagalna spremenljivka" ker pomaga razlagati program. Tako postane tisti komentar "// turn the LED on (HIGH is the voltage level)" nepotreben, saj sama koda pove točno enako.

2.6.3 Basic testing in Ardublockly

1. Connect the Arduino Uno to PC with proper USB cable.

   [Arduino Uno]

2. Run Ardublockly program. Which will be running as localhost and you will be using internet browser as IDE. The addres will be:

   http://localhost:8000/ardublockly/index.html

3. In the left corner of the program you can find [=] menu icon. From where you can choose (Slide 2 and 3)

   [] Settings:

   1. Compiler Location: C:\Program Files (x86)\Arduino\arduino_debug.exe
   2. Arduino Board: Uno
   3. Com port: COM3
   4. And press:[RETURN]

4. Finaly you can press button PLAY And if uploading was successful you will be prompted with the text (Slide 4):
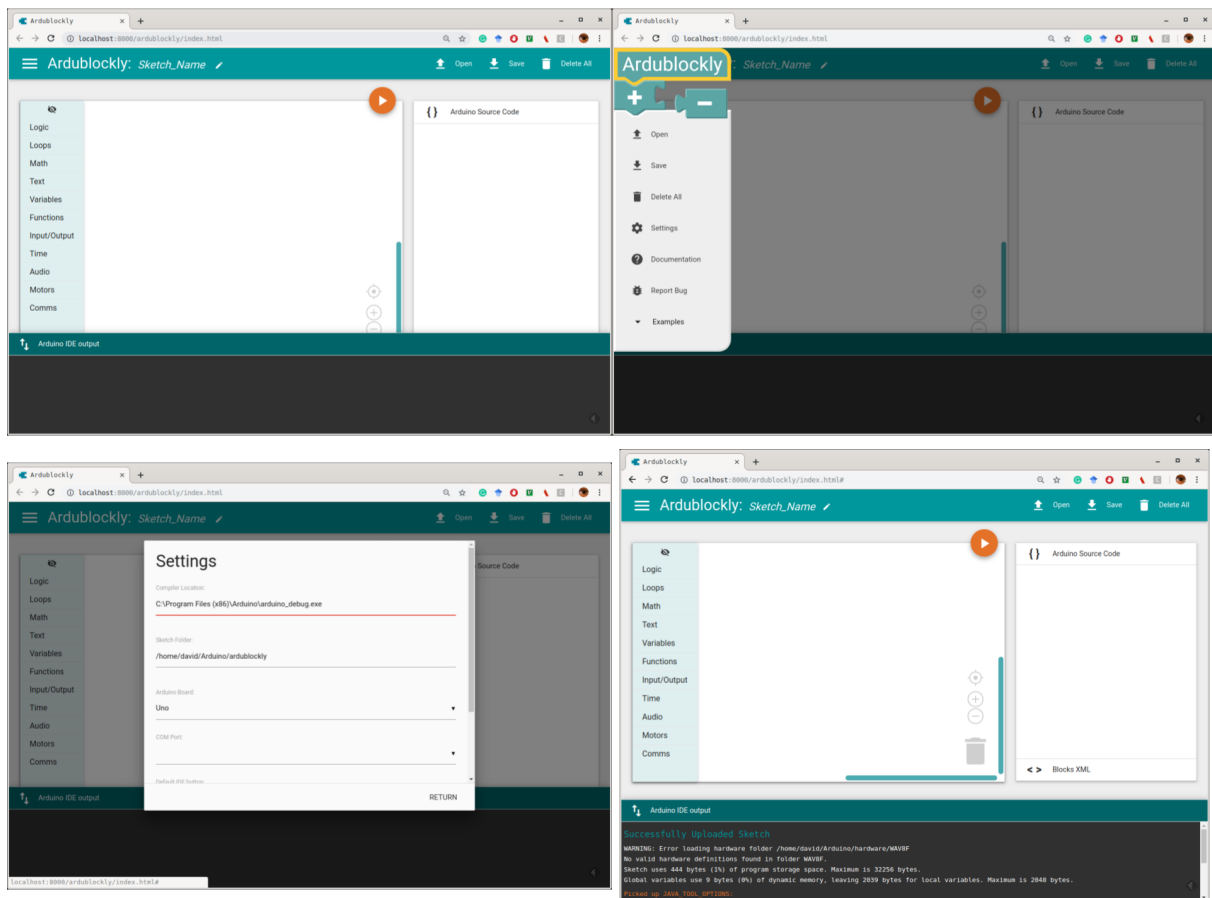
Figure 3: Ardublockly basic setup.

```
1   Successfully Uploaded Sketch
2   WARNING: Error loading hardware folder /home/david/Arduino/hardware/
        WAV8F.
3   No valid hardware definitions found in folder WAV8F.
4   Sketch uses 444 bytes (1%) of program storage space. Maximum is
5   32256 bytes. Global variables use 9 bytes (0%) of dynamic memory,
6   leaving 2039 bytes for local variables. Maximum is 2048 bytes.
```

2.6.4 Summary

Before uploading the programming code always check that the right board and serial port are set.

2.6.5 Issues

Ardublockly returns the Error id 55: Serial port Serial Port unavailable.

Try to re-connect the Arduino board. Wait a moment, check the settings and choose the COM port again then try again.

2.6.6 RobDuino module

1. Na krmilnik Arduino Uno priključite modul RobDuino in naložite naslednji program:

```
1   void setup() {
2     for (int i = 0; i < 8; i++){
3       pinMode(i, OUTPUT);
4     }
5     pinMode(A4, INPUT_PULLUP);
6     pinMode(A5, INPUT_PULLUP);
7     PORTD=1;
8   }
9
10  int l=1;
11  void loop() {
12    char tipka_a4_is_pressed = !digitalRead(A4);
13    char tipka_a5_is_pressed = !digitalRead(A5);
14    if (tipka_a4_is_pressed) l = l << 1;
15    if (tipka_a5_is_pressed) l = l >> 1;
16    if (l < 1) l = 128;
17    if (l > 255) l = 1;
18    PORTD = l;
19    delay(100);
20  }
```

2. Nato preverite delovanje obeh tipk (A4 in A5) na modulu in vrednosti izhodnih priključkov D0 .. D7.

2.6.7 Napajalni modul

Napajalni modul uporablja 2x Li-ion akumulatorja tipa 18650. Spodnje tiskano vezje je prikazano fig. 4.
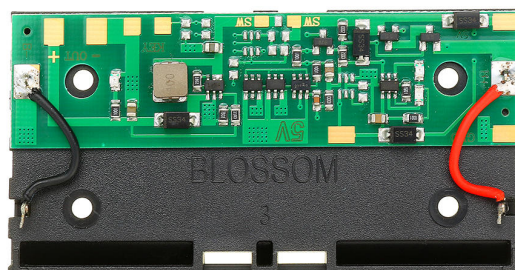
Figure 4: Napajalni modul.

Dodatno smo ga opremili z:

1. 2.5mm jack priključkom za napajanje,
2. 3-pinskim priključkom za napajanje,
3. preklopnim stikalom za izbiranje načina delovanja:

    1. ON - izhod za 9V je kaktiviran
    2. OFF - izključen izhod 9V napajanja in omogočeno je polnenje akumulatorjev preko 3-pinskega priključka (5V).

Pomembno: Pred prvo uporabo moramo ročno aktivirati napajalni modul tako, da povežemo GND na 3-pinskem priključku in NEGATIVNI terminal akumulatorjev.

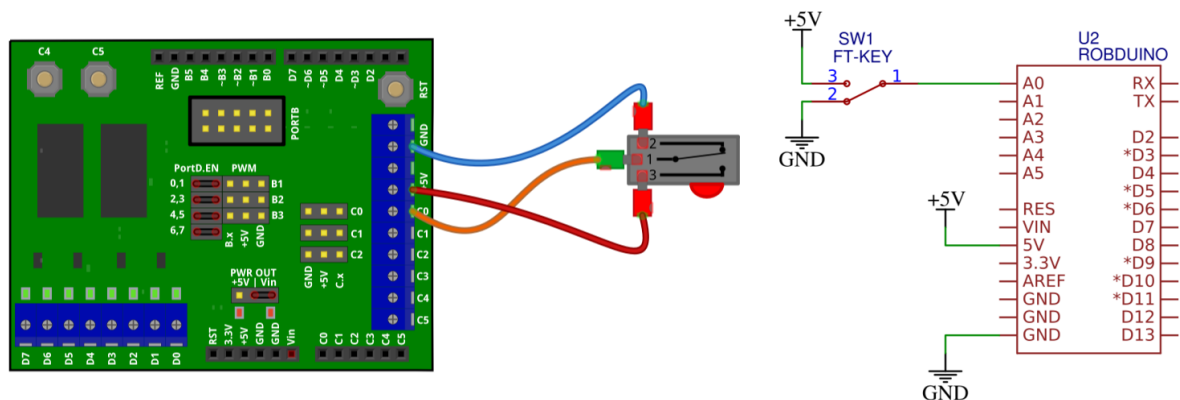### 2.6.8 Tipka

1. Priključite stikalo po shemi na fig. 5.

Figure 5: Priključitev tipke.

2. Nato naložite naslednji program.

```
void setup() {
    pinMode(A0, INPUT);
    pinMode(7, OUTPUT);
}

void loop() {
    char key_a0_is_pressed = digitalRead(A0);
    if (key_a0_is_pressed){
        digitalWrite(7, HIGH);
    } else{
        digitalWrite(7, LOW);
    }
    delay(100);
}
```

2.6.9 Svetlobni senzor

1. Priključite foto-tranzistor v delilnik napetosti z uporom, kot prikazuje fig. 6.
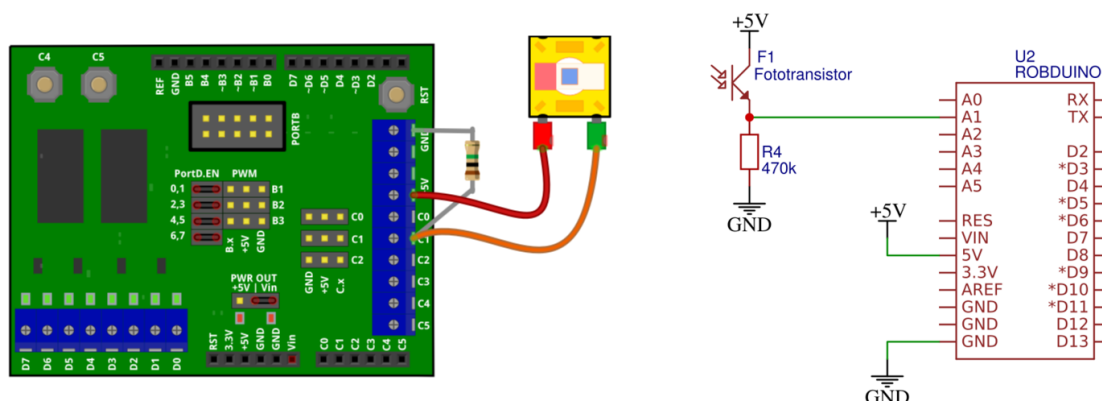
Figure 6: Priključitev foto-tranzistorja kot svetlobni senzor.

2. Nato naložite naslednji program in preverite odziv svetlobnega senzorja.

```
1   void setup() {
2     pinMode(A1, INPUT);
3     Serial.begin(9600);
4   }
5
6   void loop() {
7     int light_senzor_value = analogRead(A1);
8     Serial.println(light_senzor_value);
9     delay(100);
10  }
```

3. Odziv senzorja spremljajte v oknu serijske komunikacije.

2.6.10 IR senzor razdalje

1. IR senzor razdalje priključite na tri-pinski priključek kot je prikazano na fig. 7.
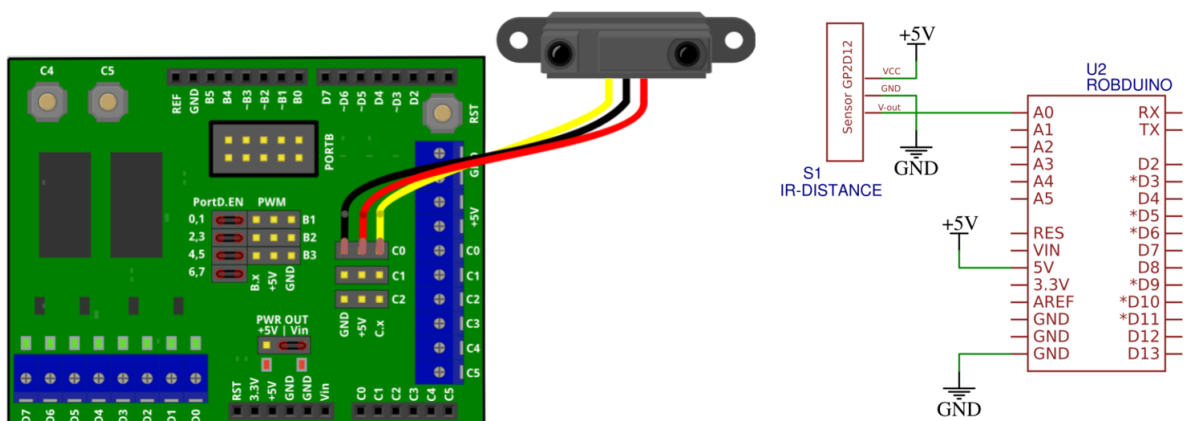
Figure 7: Priključitev IR senzorja razdalje.

2. Delovanje senzorja preskusite z naslednjim programom, njegov odziv pa spremljajte v oknu za serijsko komunikacijo.

```
1   void setup() {
2     pinMode(A0, INPUT);
3     Serial.begin(9600);
4   }
5
6   void loop() {
7     int distance_senzor_value = analogRead(A0);
8     Serial.println(distance_senzor_value);
9     delay(100);
10  }
```

2.6.11 LCD (I2C)

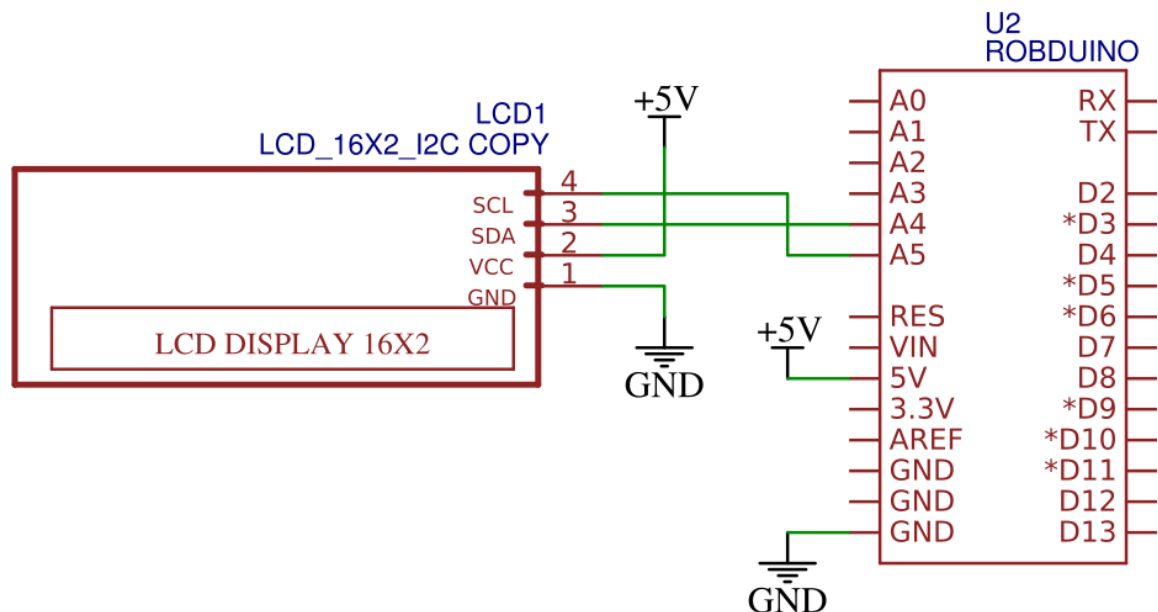1. Priključite LCD na I2C vodilo kot prikazuje

Figure 8: Povezava LCD na I2C vodilo krmilnika.

2. Priskrbite si knjižnico `LiquidCristal-I2C` iz naslova:
   https://www.arduino.cc/reference/en/libraries/liquidcrystal-i2c/

3. Knjižnico dodajte v Arduino IDE okolje tako, da dodate `ZIP` datoteko v :
   `Sketch >> Include Library >> Add .ZIP Library`

4. V VSC in PlatformIO vtičniku si lahko knjižnico naložite tako, da v terminalno okno vpišete ukaz
   `pio lib install "marcoschwartz/LiquidCrystal_I2C@^1.1.4"`

5. Nato preskusite naslednji program:

```
1    #include <Wire.h>
2    #include <LiquidCrystal_I2C.h>
3    LiquidCrystal_I2C Lcd(0x27, 16, 2);
4
5    void setup() {
6        Lcd.init();
7
8        Lcd.clear();
9        Lcd.backlight();
10
11       Lcd.setCursor(3,0);
12       Lcd.print("Hello");
13       Lcd.setCursor(6,1);
14       Lcd.print("World");
15   }
16
17   void loop() {
18   }
```

Če niste prepričani kateri i2c naslov uporablja naprava na LCD-ju le tega lahko preverite s programom I2C scanner (https://playground.arduino.cc/Main/I2cScanner/). Običajno I2C LCD-ji, ki jih naredijo kitajski proizvajalci uporabljajo I2C naslov `0x27`, `0x3F` ali manj pogosto `0x38`.