

Spiderino – a low cost robot for swarm research and educational purposes

Technical Documentation

Midhat Jdeed, Arthur Pitman, Wilfried Elmenreich

Institute of Networked and Embedded Systems

Alpen-Adria-Universität Klagenfurt, Austria

{midhat.jdeed, arthur pitman, wilfried.elmenreich}@aau.at

Version 0.3

19.02.2018

Contents

Introduction	3
1- Robot Assembling	3
2- Arduino Pro Mini 3.3V Programming	6
3- Spiderino characteristics	10
3.1 Spiderino Consumption	10
3.2 Mode Detection	11
4- Sensors	12
Outlook and Conclusion.....	14

Introduction

This report presents a design for a low-cost research robot based on the small size of the Hexbug Spider toy. Our basic modification replaces the robot head with a 3D printed adapter, consisting of two parts to provide space for sensors, a larger battery instead of the Hexbug spider's battery, and a printed circuit board (PCB) with Arduino microcontroller, Wi-Fi module, and motor controller. We address the assembling process of such a robot and the programming using Arduino studio. The presented prototype costs less than 70 Euro, and is suitable for swarm robotic experiments and educational purposes [1].

1- Robot Assembling

Basically, assembling Spiderino includes three basic steps:

- 1- Preparing and adjusting a Hexbug Spider to fit in the proposed robot.
- 2- Printing 3D adapter parts for the sensors, battery and PCB.
- 3- Creating a circuit board by soldering and connecting electronic components over it.

The proposed robot design consists of the locomotion system of a Hexbug Spider where we have attached a 3D-printed adapter. The physical parameters derive mainly from the Hexbug Spider which has six legs. The mechanics provide a coordinated movement for all six legs to move the robot. To change the direction, the robot has to turn its head.

The Hexbug Spider has as mechanical movement system with six legs and two small electric engines for simultaneous, coordinated movement of the spider legs. One engine is used for rotary motion, the second for forward or backward movement. The goal of the mechanical design of the robot is to yield mechanical robustness and reliability of the movement, as well as for low energy consumption and low cost of overall design, aiming at light weight mechanical variants.

For the current prototype, our modification starts with partly disassembling the original robot. The head of the spider needs to be removed in order to get access to the motors. As replacement of the head, 3D-printed adapter parts are attached to the robot. Figure 1 shows Hexbug Spider before and after the disassembling step.

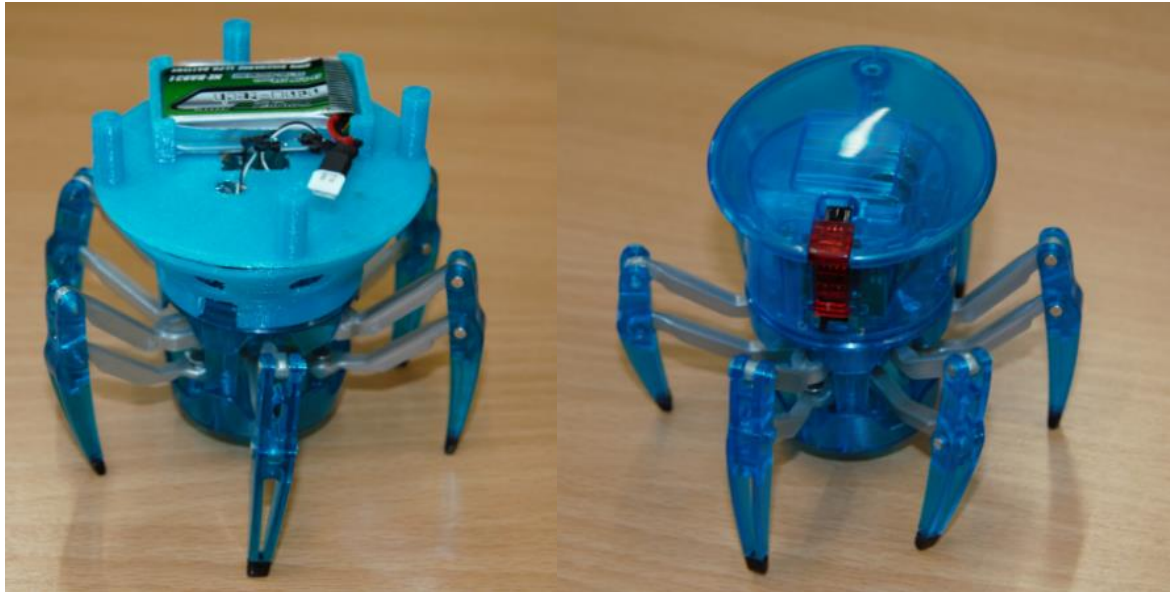


Figure 1. Replacing the Hexbug Spider's head with a 3D printed adapter

Regarding the third step, we used an Arduino Pro Mini with an ATmega328 processor (3.3V, 8 MHz), Wi-Fi module ESP8266, and POLOLU_Motor_DRV_DRV8835 to control the two engines of the Hexbug Spider. Table 1 shows the other components we connected on the PCB (Figure 2) for one Spidrobot.

Element	Number	Library
Battery nano-tech	1	Pinhead
Resistance 560K	1	Rcl
Resistance 120	2	Rcl
Resistance 10K	1	Rcl
Resistance 15K	1	Rcl
LED	2	Led
Power switch	1	Switch

Table 1. PCB components

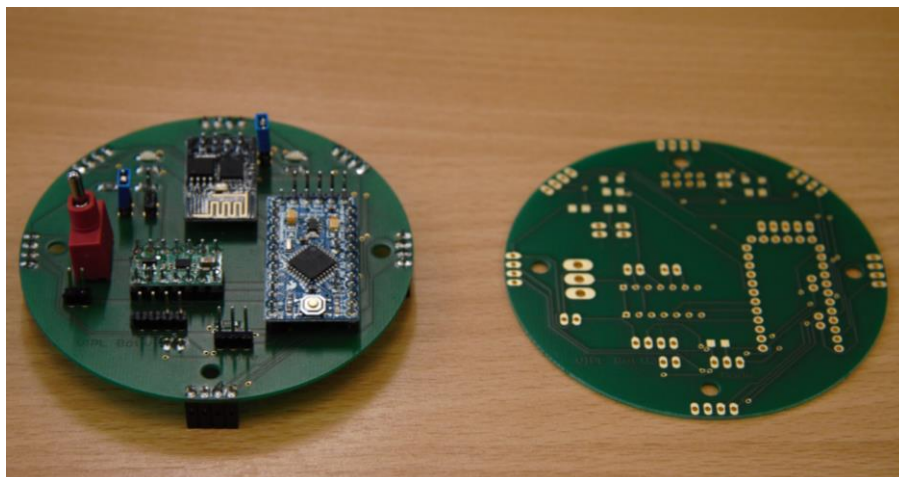


Figure 2. PCB with components

Before uploading a test program (an empty sketch) to the Arduino Pro Mini, ensure that everything is connected and soldered well on the PCB. Figure 3 shows the current prototype after assembling the parts together: the Hexbug Spider, the battery, the 3D printed adapter and the PCB. We prepared a swarm of ten Spiderbots as it is also shown in Figure 3.

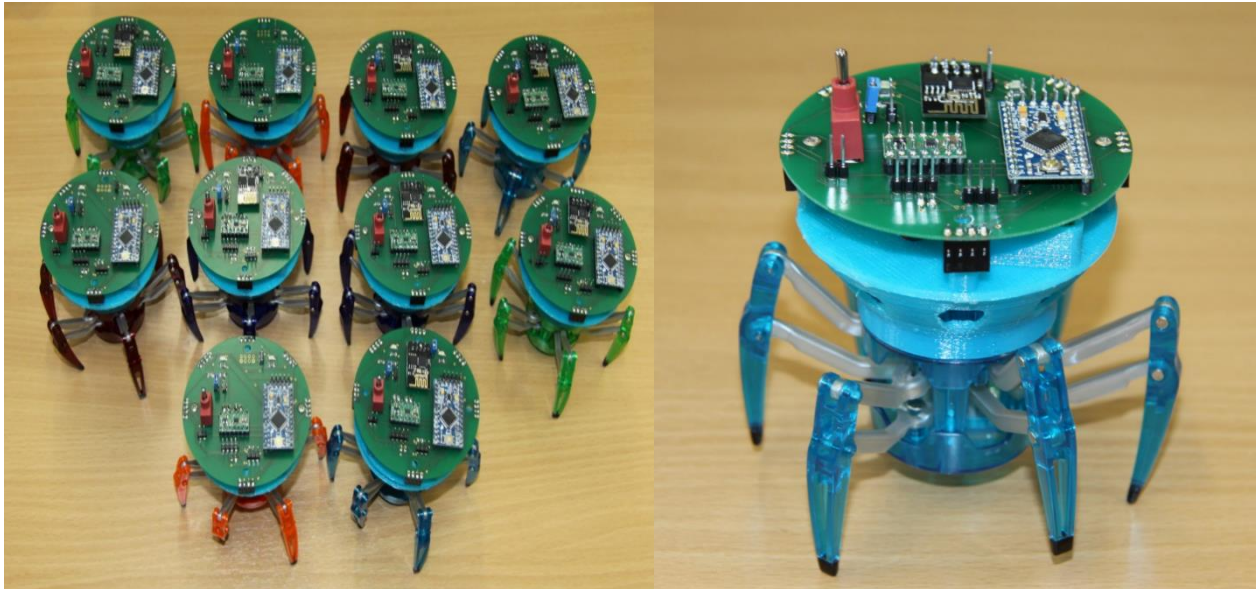


Figure 3. Spiderinos

The PCB is fixed on the top of the adapter using two screws. It could even be fixed using four screws according to the printed adapter structure.

Basically, the main properties for our robot prototype are [2]:

- Affordable. The total price of one robot including additional modules should not exceed 100 euro. The body of the robot can be easily reproduced using a 3D printer.
- Swarm-oriented. The robots will be used in experiments with swarms and the components of the robot should enhance cooperation between robots.
- Customizable. The model can be changed in order to meet the requirements, for example, to add additional sensors.
- Open-platform. All models and blueprints of the robot are a freely available information and everyone can use, reproduce or modify them.
- Easy to use. Provide simple programming and user friendly robotic implementation.

2- Arduino Pro Mini 3.3V Programming

On the PCB we added a connector for serial programming to the Arduino Pro Mini with the following wiring scheme, Figure 4:

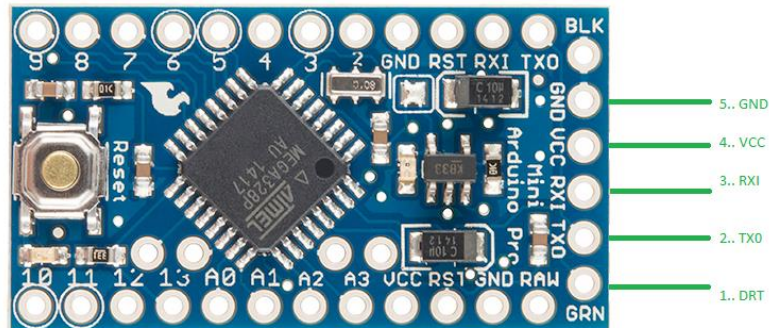


Figure 4. Wiring schema for Arduino Pro Mini

We used a BTE13-007 as a programmer where it has a USB connector which maps itself as a COM port. BTE13-007 has to be connected with the PCB, this can be achieved with a five-wire connector (the CTS of BTE13-007 will not be used) as shown in Figure 5.

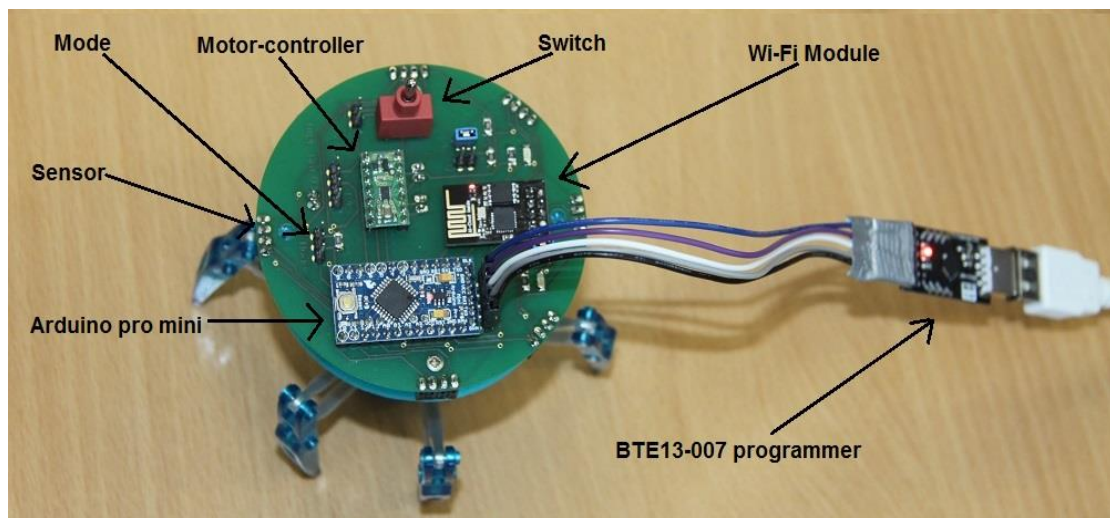


Figure 5. Connecting BTE13-007 to the PCB

According to Figure 5, we used the following colour schemes:

- Blue – GND – Pin5 (GND).
- Violet – 5V – Pin4 (VCC).
- White – TXD – Pin3 (RXI).
- Gray – RXD – Pin2 (TXO).
- Black – DTR – Pin1 (DTR).

Figure 6 shows the correct COM-port settings using Arduino Studio, also the other settings for the Arduino type microcontroller and the programmer type.

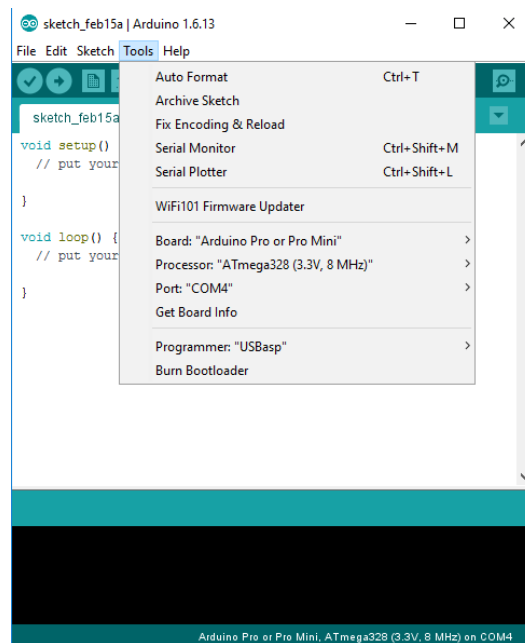


Figure 6. Arduino Studio settings

Then, we have programmed Arduino with basic functions illustrated in Table 2. The integers in the code are indicating the pin numbers on Arduino Pro Mini which is used to control its target. For example, on the PCB pin numbers 7 and 5 are connecting to one engine of the Hexbug Spider which is responsible for walking forward and backward.

Functionality	The Code
Walking forward	<pre>void walkForward () { digitalWrite (7, LOW); pinMode (7, OUTPUT); digitalWrite (5, HIGH); pinMode (5, OUTPUT); }</pre>
Walking backward	<pre>void walkBackward () { digitalWrite (7, HIGH); pinMode (7, OUTPUT); digitalWrite (5, HIGH); pinMode (5, OUTPUT); }</pre>
Stop walking	<pre>void walkStop () { digitalWrite (5, LOW); pinMode (5, OUTPUT); }</pre>
Turning right	<pre>void turnRight () {</pre>

	<pre>digitalWrite (6, HIGH); pinMode (6, OUTPUT); digitalWrite (4, HIGH); pinMode (4, OUTPUT); }</pre>
Turning left	<pre>void turnLeft () { digitalWrite (6, HIGH); pinMode (6, OUTPUT); digitalWrite (4, LOW); pinMode (4, OUTPUT); }</pre>
Stop turning	<pre>void turnStop () { digitalWrite (6, LOW); pinMode (6, OUTPUT); }</pre>
Turn on the right LED	<pre>void rightLED () { pinMode (2, OUTPUT); digitalWrite (2, LOW); }</pre>
Turn on the left LED	<pre>void leftLED () { pinMode (3, OUTPUT); digitalWrite (3, LOW); }</pre>
Turn off the right LED	<pre>void rightLEDOff () { pinMode (2, INPUT); digitalWrite (2, LOW); }</pre>
Turn off the left LED	<pre>void leftLEDOff () { pinMode (3, INPUT); digitalWrite (3, LOW); }</pre>
Pullup-on the internal resistance	<pre>void pullUpOn () { pinMode (9, INPUT_PULLUP); digitalWrite (9, HIGH); }</pre>
Pullup-off the internal resistance	<pre>void pullUpOff () { pinMode (9, INPUT_PULLUP); digitalWrite (9, LOW); }</pre>
Pin number 9 -modes	<pre>void getMode () { pullUpOn (); if (digitalRead (9) == HIGH) { { pullUpOff (); if (digitalRead(9) == LOW) { return 1; } } else {</pre>

	<pre> return 2; } } } else if (digitalRead (9) == LOW) { return 3; } } </pre>
--	---

Table 2. The basic functions

In order to test and set up the Wi-Fi module, we have sent *AT* commands from a computer via an USB to serial adapter. In the proposed prototype, it is wired with Arduino Pro Mini on the PCB as following Table 3:

Wi-Fi ESP8266	Arduino Pro Mini
VCC - must be no more than 3.6V	VCC
GND	GND
TX	RX1
RX	TX0
CH-PD > VCC	---

Table 3. Connecting ESP8266 with Arduino Pro Mini

Therefore, to send *AT* commands using BTE13-007, we need to swap the connection between BTE13-007 and Arduino Pro Mini as following:

- **White – TXD – Pin2 (TX0).**
- **Gray – RXD – Pin3 (RX1).**

With that connection, we will not be able to program Arduino Pro Mini any more, it is just to test the Wi-Fi module on the PCB using serial monitor in Arduino studio as it is shown in Figure 7. We ensured *AT* commands are received correctly, for example, we enable the Wi-Fi module to act as both a “Station” and an “Access Point” using *AT+CWMODE=3* command.

Note: In order to send *AT* commands from Arduino Pro Mini **8 MHz** to Wi-Fi module ESP8266, we need to ensure that the baud rate in ESP8266 is 9600 or 57600, because the Arduino with 8Mhz processor does not work correctly with high baud rate like 115200. ESP8266 baud speed can be changed by sending the **AT+CI0BAUD** command with the required value.

We can also connect the Wi-Fi module directly with BTE13-007 and test or even program it using Arduino studio. The instructions for that case are not included in this report.



```
COM4
AT
OK
AT+GMR
AT version:0.40.0.0(Aug 8 2015 14:45:58)
SDK version:1.3.0
AI-Thinker Technology Co.,Ltd.
Build:1.3.0.2 Sep 11 2015 11:48:04
OK
AT+CMODE=3
OK
```

Figure 7. AT commands using Serial monitor

3- Spiderino characteristics

Using Arduino Studio allows us to program Arduino microcontroller in order to control the two engines of Spiderino and implement the basic functionalities such as walking, turning and lighting the two LEDs. In addition, there is a software library written in C/C++ programming languages to control the motor speed and read information from the proximity sensors. This library can be easily imported into Arduino Studio to implement a firmware for Spiderino. We will use it in the current prototype after adding five distance sensors to Spiderino. However, for the test cases, we programmed Arduino microcontroller with the basic functionalities in order to measure speed and power consumption and to test the mode switch functionality.

3.1 Spiderino Consumption

Spiderino prototype has a LiPo battery with a capacity of 750mAh. A simple test program was uploaded that makes Spiderino walk, turn and light two LEDs in order to measure the power consumption. Figure 8 illustrates the measurements for these functionalities. The Autonomy of Spiderino is between 4 and 20 hours where it depends on the number of modules used in an experiment. For example, in case Spiderino is only walking on a wood table, the battery could power the Spiderino for 15 hours. However, if Spiderino is walking on a carpet that is not smoothed well, then its consumption will increase, and the maximum consumption can then reach 60mA. Additionally, Spiderino can move with a speed of 6 cm/sec. One moving cycle of all six legs takes 500 milliseconds. Regarding head turning, Spiderino has to turn its head in order to change its direction, a full turn takes 3 seconds.

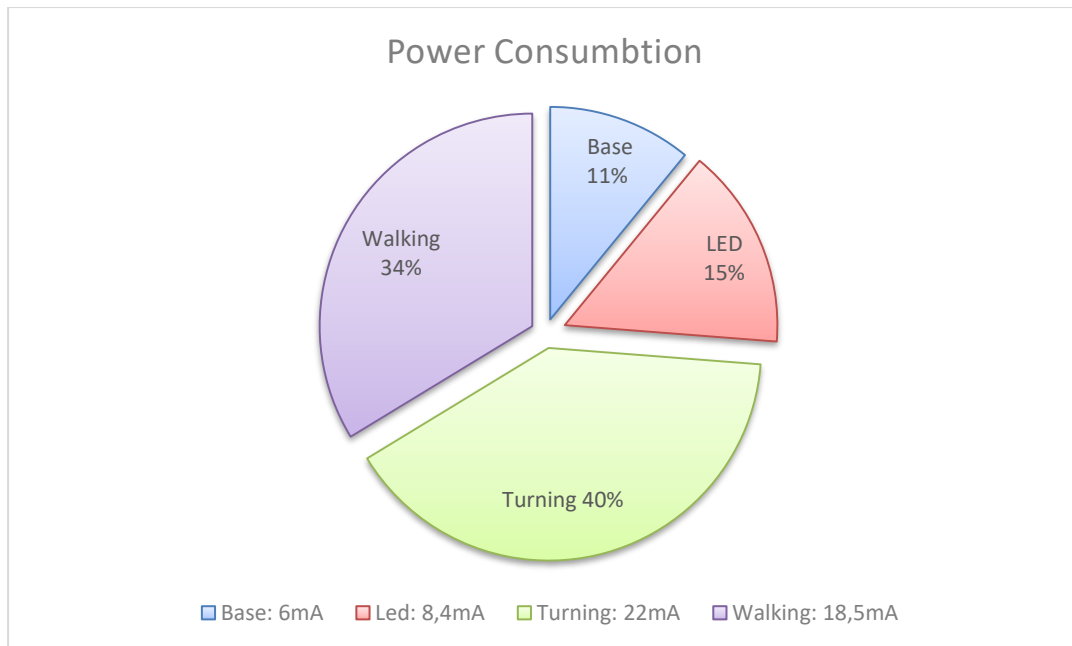


Figure 8. Power Consumption

The ESP8266 Wi-Fi module, it has a standard rate for power consumption according to different modes included in the ESP8266 datasheet. During the communication between Wi-Fi module and Arduino pro mini in order to test the connection, the regulated power consumption was around 70mA. Usually, the module will be in deep sleep mode in order to save power¹.

3.2 Mode Detection

Arduino Pro Mini platform has an internal pull-up resistor of 20-50 kOhms that can be enabled or disabled to a specific digital pin on Arduino². We used this feature in order to achieve three mode possibilities for a signal digital input pin. The designed PCB has a jumper with three pins as shown in Figure 1 providing three switching possibilities which are no jumper, a jumper to the right side or to the left side.

Arduino Pro Mini has 14 digital pins and they can be used as an input, output or INPUT-PULLUP using `pinMode()`, `digitalWrite()`, and `digitalRead()` functions. They operate at 3.3V in our case. The internal pull-up resistor can be activated using `pinMode()` function as follows:

```
pinMode(9, INPUT-PULLUP); // Enable internal pull-up resistor on pin 9
```

On the PCB, we added an external pull-up resistor in order to obtain three modes regarding pin number 9 in Arduino Pro Mini. Table 4 illustrates these possibilities or modes. The external pull-up resistor value should consider two conditions:

¹ <https://nurdspace.nl/ESP8266>

² <https://www.arduino.cc/en/Main/arduinoBoardProMini>

- 1- Voltage divider equation for two resistors as it is shown in the following equation:

$$V_{out} = V_{in} * R_2 / (R_1 + R_2)$$

where: R1, R2 are the internal and external resistors respectively.

- 2- digitalRead() function will read a pin voltage to the corresponding value: 3.3V for HIGH, 0V (ground) for LOW.

External pull-up	Internal pull-up	RETURN
LOW	HIGH	Mode1
HIGH	HIGH	Mode2
LOW	LOW	Mode3

Table 4: Spiderino Modes

4- Sensors

The used sensor for Spiderino is CNY70³. Five sensors can be added for each Spiderino. CNY70 can measure distances and detect obstacles based on the idea that the received light decreases with increased distance. Figure 9 illustrates the wiring schema for CNY70 sensor on Spiderino board.

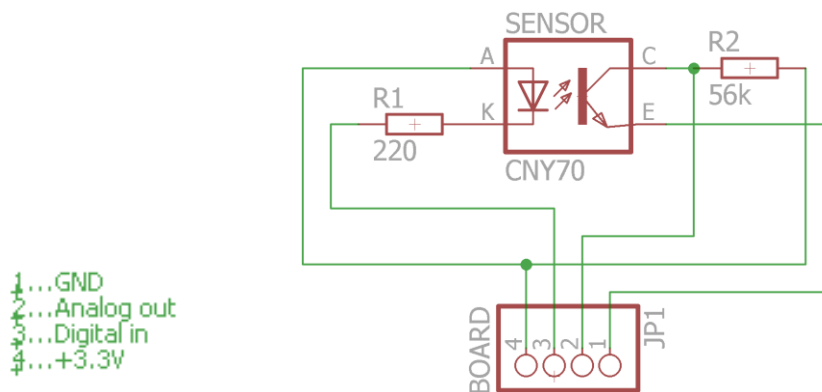


Figure 9. CNY70 sensor wiring schema

Each two sensors have the same digital pin on Arduino Pro Mini, and each sensor has one Analog pin number. The used digital pins are 11, 12 and 13. The Analog ones are A0, A1, A3, A6 and A7. The following code illustrates how we can read the sensor data using the Serial Monitor in Arduino Studio. By turning on and off the infrared LEDs, a Spiderino may distinguish between light sources and obstacles, such as other Spiderinos. Empirical evidence from an investigation into the sensor properties suggest that a Spiderino may detect obstacles within 0.025m and light sources within ranges as diverse as 0.2m or even 2m, depending on light intensity.

³ <https://www.vishay.com/docs/83751/cny70.pdf>

```

Int sensor1;
void setup()
{
  Serial.begin(9600);
  pinMode(11, OUTPUT); \\ 12,13
}

void loop()
{
  Sensor1 =analogRead (A1); \\ A0, A2, A3, A6, A7
  Serial.println(Sensor1);
}

```

Table 5. Testing sensors value using Arduino studio

Moreover, Spiderino has a distance measuring sensor (GP2D12), see Figure 10, with an effective range of 0.8 m. This sensor is connected to the analog pin (A2) on the Arduino Pro Mini. To test the sensor values using the serial monitor in Arduino studio for the GP2D12 or CNY70, see Table 5.

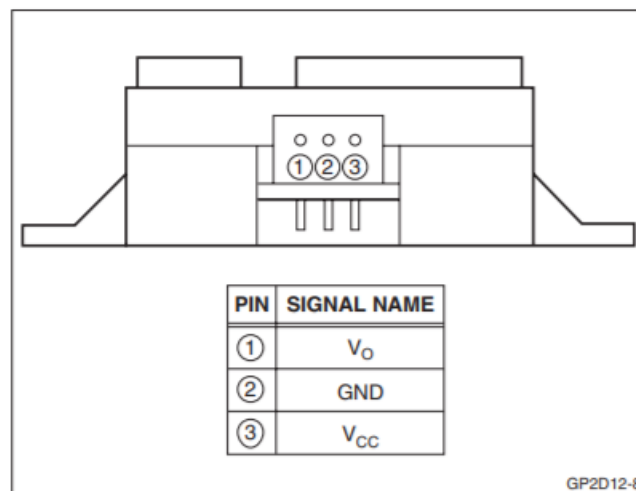


Figure 10. GP2D12 Pinout⁴

⁴ https://engineering.purdue.edu/ME588/SpecSheets/sharp_gp2d12.pdf

Outlook and Conclusion

In this report, we presented a robot design 'Spiderino' which is an extension of the Hexbug Spider toy robot. The overall assembly of the final robot is quick and the material costs are low.

In our 'to do' list, simulating a swarm behavior using ten Spiderinos. For that purpose, we will use FREVO tool. In addition, the swarm of ten Spiderinos will be used to develop and evaluate a swarm algorithm.

References

- [1] Jdeed, M., Zhevzyk, S., Steinkellner, F., & Elmenreich, W. (2017, June). Spiderino-A low-cost robot for swarm research and educational purposes. In *Intelligent Solutions in Embedded Systems (WISES), 2017 13th Workshop on*(pp. 35-39). IEEE.
- [2] Elmenreich, W., Heiden, B., Reiner, G., & Zhevzyk, S. (2015, October). A low-cost robot for multi-robot experiments. In *Intelligent Solutions in Embedded Systems (WISES), 2015 12th International Workshop on* (pp. 127-132). IEEE.