A ROS2-based Swarm Robot Platform: SMARTmBOT

Jaeeun Kim, Wonse Jo, and Byung-Cheol Min

***Abstract*— This paper is to introduce a new open-source swarm robot platform for a multi-robot research applications. The mobile robot is 15 cm in diameter and 10 cm in height and supports the Robot Operating System 2 (ROS2). The hardware is composed of three layers of printed circuit board (PCB). The first PCB layer has two DC motors with a motor driver chip (L293D), and two line sensors to detect cliffs or black lines. The second PCB layer is to connect with a Raspberry Pi and eight times-of-flight (ToF) sensors to detect surrounding obstacles. The third PCB layer is customizable. In our case, we mounted four RGB SMD LEDs (SMD5050), RGB LED strips (WS2913b), and a speaker on the thrid PCB layer. Therefore, the robot platform is able to change its light either individually or as a whole, play music, and measure the distance of nearby obstacles using ToF sensors. Furthermore, we provide all source codes and design files used on the mobile robot platform.**

I. INTRODUCTION

Swarm robotics is a potential field that can solve complex and difficult challenges in robotics applications. Various robot platforms have been introduced by companies and research groups for swarm robotics.

To address these challenges, various research institutes, universities, and companies have been launching various STEM tools. An example of the mobile robot platforms is Donkey Car, which provides the learning tools for driving an RC car and process data collected with it [1]. Similarly, Miniskybot aims to be a learning tool that is cheap and easy to build [2]. It has 3D printed parts and its own processing board, which is controlled by a personal computer. Although there are aforementioned mobile platforms that allow the user to build from scratch and modify as they need, there is still a huge barrier for students to learn STEM educations, as well as for the researchers who are not in robotics to conduct human-robot interaction (HRI).

However, those platforms are still expensive and limited in expandability for other research to use.

Therefore, we propose a new fully open-source swarm robot platform, called SMARTmBOT that can be used for swarm research purposes. The SARTmBOT is made with a 3D printed casing and using a Raspberry Pi 4 installed Robot Operating System 2 (ROS2) [3] to control the devices and read sensors mounted on three layers of the printed circuit board (PCB). The sensors and visual output components are mounted on the PCBs, such as RGB LEDs, Time of Flight (ToF) sensors, photo reflector sensors (also known as a line sensor), and DC gear motors. The power source for SMARTmBOT is a portable power bank, and the rest of the sensors and components are powered by the Raspberry Pi 4. As well, we elaborated specific details of software in Section III.

Jaeeun Kim, Wonse Jo, and Byung-Cheol Min are with SMART Lab, Department of Computer and Information Technology, Purdue University, West Lafayette, IN 47907, USA `kim2592@purdue.edu`, `jow@purdue.edu`, `minb@purdue.edu`

Fig. 1: SMARTmBOT platforms mounted a WS2813B RGB LED strip to change the overall case.

II. HARDWARE

The components used in the SMARTmBOT are shown in Table I, composed of off-the-shelf items. The cost for building a SMARTmBOT is about $ 250. Furthermore, in order to assist those who are building the robot, a video to introduce how to solder and assembly the platform is available. To find the video, please go to this link: `https://youtu.be/hEa0x9cL72s`

A. 3D designs

The SMARTmBOT is 15 cm in diameter and 10 cm in height, and composed of 3D printed parts as shown in Figure 3. All parts should be completely printed and soldered before assembling them. Figure 3a is a transparent case to cover the whole robot platform and illuminate color lights using RGB LEDs. Figure 3b is a power bank holder that serves as the main base of the SMRTmBOT. The part is to mount the power bank and connect other PCB layers and Raspberry Pi 4. Figure 3c is a motor mount bracket to fix two DC motors. The perforated hole is to pass and arrange a power cable and cables of the DC motors. Figure 3d is a shaft support bracket to fasten the the shaft of belt chain on the power bank holder. Figure 3e is a motor mount bracket to fix a Pi camera

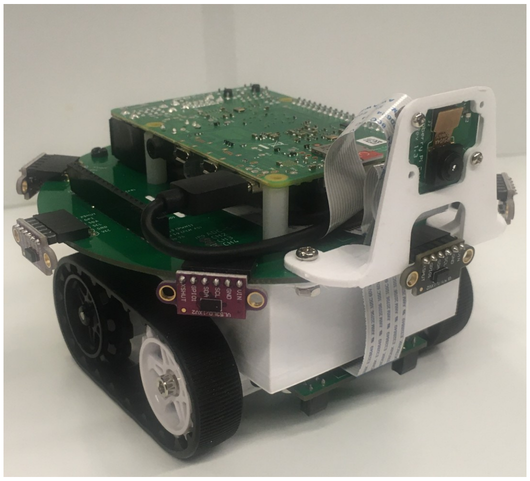


Fig. 2: Basic version of the SMARTmBOT platform.

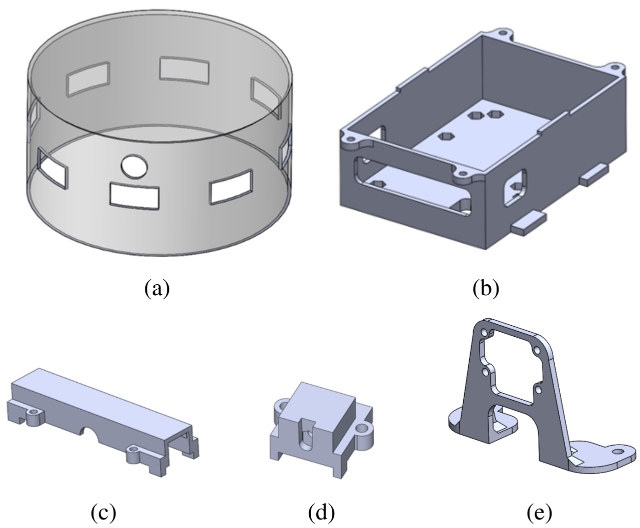


Fig. 3: 3D printing parts for SMARTmBOT; (a) Transparent case (1EA), (b)Power bank holder, (c) Motor mount bracket, (d) Shaft support bracket, and (e) Camera holder.

on the second PCB layer. All design and 3D printing files can be downloaded from: `https://github.com/SMARTlab-Purdue/SMARTmBOT/tree/main/Hardware_design/3DprintingParts`.

B. PCB design

1) First Layer: The first PCB layer shown in Figure 4a is located between the wheels, facing downwards. It includes a motor driver (L293D), ADC-SPI chip (MCP3008), and two photo reflector sensors (RPR220). Motor lines and external power supply from the battery is connected to this layer.

2) Second Layer: The second layer is placed on top of the battery housing, holding mounts for Raspberry Pi and ToF sensors.

3) Third Layer: The third layer is placed on top of the second layer, connected by pin headers.

4) Custom Layer: This layer provides extra pins on the third layer PCB to connect an additional layers.

TABLE I: BOM list for the SMARTmBOT

#	Item	Quantity	Price
1	Power bank	1	$ 22.99
2	DC gear Motor	2	$ 17.95
3	32GB SD card	1	$ 7.49
4	Raspberry Pi4 4GB	1	$ 65.06
5	Pi camera	1	$ 7.99
6	22T track wheel set	1	$ 11.66
7	RPR220 photo reflector sensor	2	$ 1.10
8	VL53L0X time-of-flight laser ranging sensor	8	$ 7.45
9	550 SMD RGB LED	4	$ 0.53
10	MCP3008	1	$ 1.82
11	L293D	1	$ 3.32
12	2.5mm JST female and male connector	3 sets	$ 0.45
13	FFC connector	2	$ 1.28
14	FFC cable	1	$ 2.44
15	100 Ohm & 200 Ohm &68k Ohm resistor	1	$ 0.11
16	90 degree angle pin header	8	$ 0.80
17	Long 20 pin female header	2	$ 1.37
18	20 pin female and male header	2 sets	$ 0.34
19	Spacer (5mm & 100mm)	4 sets	$ 0.72
20	M3 bolts and nuts	20 sets	$ 0.2
21	PCB printing (Basement layers, 1st and 2nd)	1 set	$ 10.90

Total cost: $ 254.21 as per a robot

III. Software

The software required to run the ROS2 packages are available for download at `https://github.com/SMARTlab-Purdue/smart_mbot_ws`. The system runs on Ubuntu 20.04 Server and ROS2 Galactic.

A. Control the LEDs with GPIO Pins

The LEDs can emit red, green, or blue light. Each color is connected to a separate GPIO pin, so the user can turn individual or multiple colors at the same time. The pins used for the LEDs are 7, 8, and 25. Below is the part of LED code included in the package. To control the LED colors, set the output of the wanted color's pin to HIGH by changing the GPIO.LOW to GPIO.HIGH.

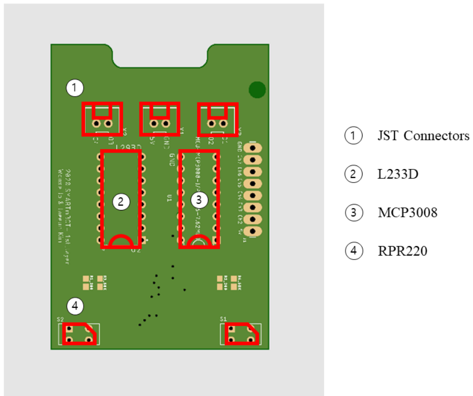
Code 1: Python example

```
GPIO.output(pin, GPIO.LOW)
```

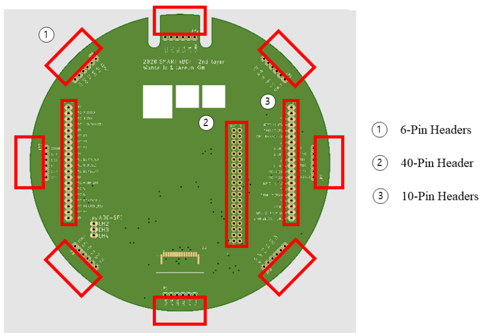
B. Read ADC via SPI Node

The photo reflector sensors used in this robot, RPR220, are analog sensors which signals are converted into digital signals and read on SPI bus via MCP3008 chip. The pin used for the sensor is 5.

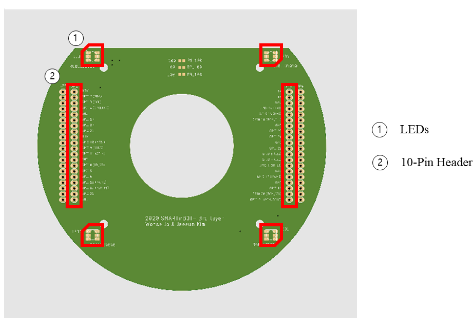
The code in the package will convert the analog data to voltage values, stored in an array. Below is the part of the



(a) First PCB layer

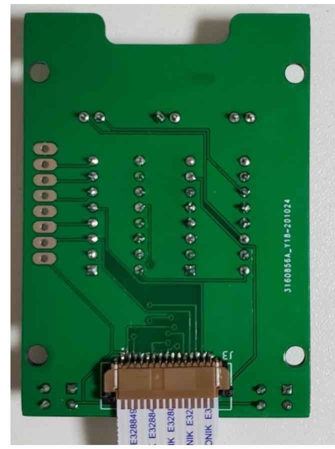


(b) Second PCB layer

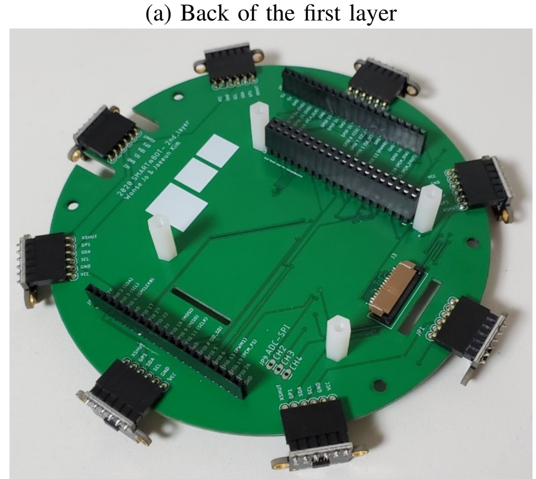


(c) Third PCB layer

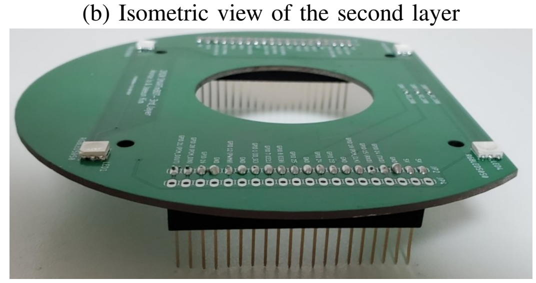
Fig. 4: Gerber design files for the PCB layers;



(a) Back of the first layer



(b) Isometric view of the second layer



(c) Isometric view of the third view

Fig. 5: Three simple graphs

code used for reading data from the sensors. The data is automatically published as ROS2 message *Float32MultiArray* as 'smartmobil/ADC'.

Code 2: Python example

```
# Initialize and define Publishers
self.pub_adc_spi_reader = self.create_publisher(
    Float32MultiArray, 'smartmobil/ADC', 10)

# Setting timer for publisher
pub_adc_spi_reader_timer_period = 0.001 #sec
self.pub_adc_spi_reader_timer =
self.create_timer(pub_adc_spi_reader_timer_period,
    self.pub_adc_spi_reader_timer_callback)
```

C. Control DC Motors

We used L293D motor driver chip to control two DC motors with pulse-width modulation (PWM). The pins used for first motor is 37, 33, 35, and second motor is 36, 38, 40. The last pins listed in each motors are used for PWM control.

The code below is displaying the equation used to calculate each wheel's velocity with given linear and angular velocity. WHEEL_BASE is the distance between the two wheels and WHEEL_RADIUS is the radius of the wheel.

Code 3: Python example

```
self.right_velocity = abs((self.lin_vel - self.
    ang_vel * WHEEL_BASE / 2.0) / WHEEL_RADIUS)
```

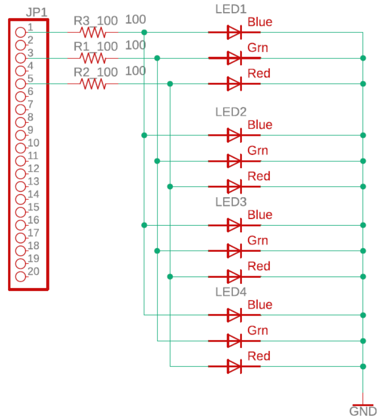


Fig. 6: Schematic of four LEDs (not an exhaustive diagram)

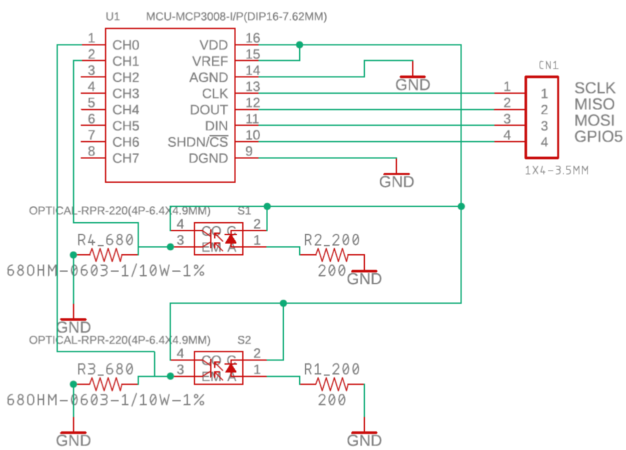


Fig. 7: Schematic of SPI-ADC wiring with photo reflector sensors (not an exhaustive diagram)

```
self.left_velocity = abs((self.lin_vel + self.
    ang_vel * WHEEL_BASE / 2.0) / WHEEL_RADIUS)
```

D. *Read ToF Sensors*

The eight ToF sensors or on a I2C bus, each with their unique I2C addresses. The pins used for the sensors are 13, 18, 16, 15, 12, 11, 8, and 10.

Below is the part of the code used for reading data from the sensors. The data is automatically published as ROS2 message Int32MultiArray as 'smartmobil/I2C'.

Code 4: Python example

```
# Initialize the I2C module
self.i2c = board.I2C()
self.pub_i2c_tof_reader = self.create_publisher(
    Int32MultiArray, 'smartmobil/I2C', 10)
```

IV. ROS 2 LAUNCH

Code 5 is a ROS2 launch file to run a group of nodes at a same time. For this application, the launch file included in the package runs motor, SPI, I^{2}C, and LED strip nodes together.

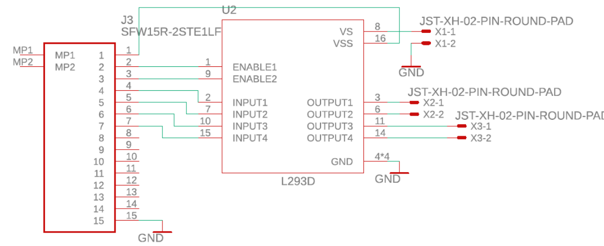


Fig. 8: Schematic of motor driver with two DC motors and an external power source (not an exhaustive diagram)

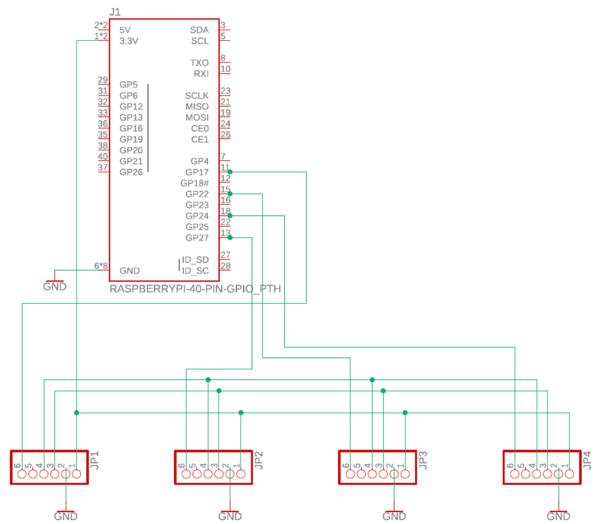


Fig. 9: Schematic of I2C wiring with four ToF sensors (not an exhaustive diagram)

Code 5: Launch file to control SMARTmBOT

```
import numpy as np
```

To launch the package, type the following commands to the terminal:

```
$cd ~/smart_mbot_ws
$colcon build
$source install/setup.bash
$ros2 launch smart_mbot_pkg $smart_mbot_pkg.launch.
    py
```

There are two subscribers and two publishers in the package. Since they are in ROS2 message formats, the user can write to the publisher and read the publishers by accessing the topics with appropriate messages. The topics are listed and explained further in Table II and Table III.

As an example, it is possible to directly publish a message

TABLE II: ROS2 Topic Subscriber List

Name	Type	Description
/writing_dc_motor_vel	*Float32MultiArray*	Linear and Angular Velocity for the Motors
/writing_ws2813_rgb_strip	*Float32MultiArray*	LED Strip Parameters

TABLE III: ROS2 Topic Publisher List

Name	Type	Description
/reading_i2c_VL53L0X_tof	*Int32MultiArray*	Array of ToF Sensor Values
/reading_spi_adc	*Float32MultiArray*	Array of Photo Reflector Sensor Values

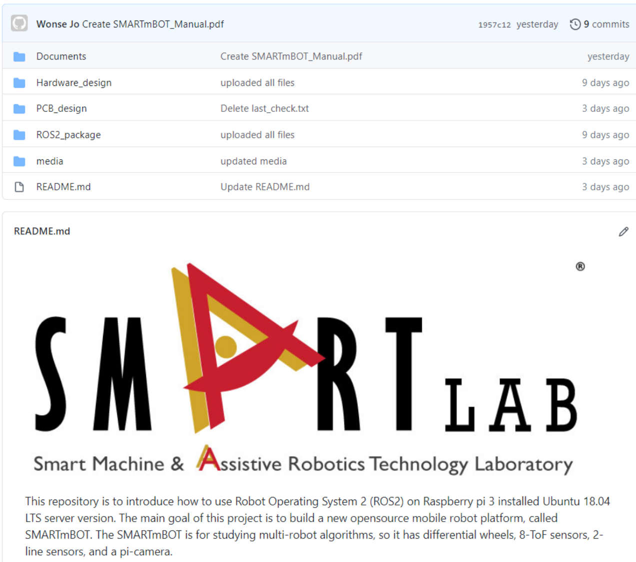


Fig. 10: SMARTmBOT GitHub page

to /writing_dc_motor_vel to control the motors. The message type Float32MultiArray accepts right and left motor velocities as a form of "data: [right_motor_pwm(-255 255), left_motor_pwm(-255 255)]". When running in the terminal, the command should look like Code 6.

Code 6: Controlling motors of SMARTmBOT

```
ros2 topic pub /<robot_name>/writing_dc_motor_vel
    std_msgs/msg/Float32MultiArray "data: [100,100]
    "
```

V. Online Repositories

The GitHub repository can be found in the following link: `https://github.com/SMARTlab-Purdue/SMARTmBOT`. In the repository, there are 3D design CAD files, PCB design files, and ROS2 package utilized on the SMARTmBOT. The 3D design files are created by SolidWorks 2020, the PCB designs are created by AutoDesk Eagle CAD, and ROS2 nodes are written using Python 3.8.

VI. Conclusion and Future works

In this paper, we presented an open-source mobile robot platform to serve as a learning tool for K-12 student. The components are relatively cheap and mainstream so that they are easy to obtain, and this manual will help the students to assemble and modify the codes.

In the future, we will conduct a new user study to investigate the effect of the multi-robot system on human perception and cognition. And also, we will share the mobile platform with K-12 students to give opportunities to learn human-robot interaction via virtual meeting and/or in-person meeting.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. IIS-1846221. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] Autorope, "donkeycar," https://github.com/autorope/donkeycar, 2020.

[2] J. Gonzalez-Gomez, A. Valero-Gomez, A. Prieto-Moreno, and M. Abderrahim, "A new open source 3d-printable mobile robotic platform for education," in *Advances in autonomous mini robots*. Springer, 2012, pp. 49–62.

[3] "Ros 2 documentation¶." [Online]. Available: https://index.ros.org/doc/ros2/