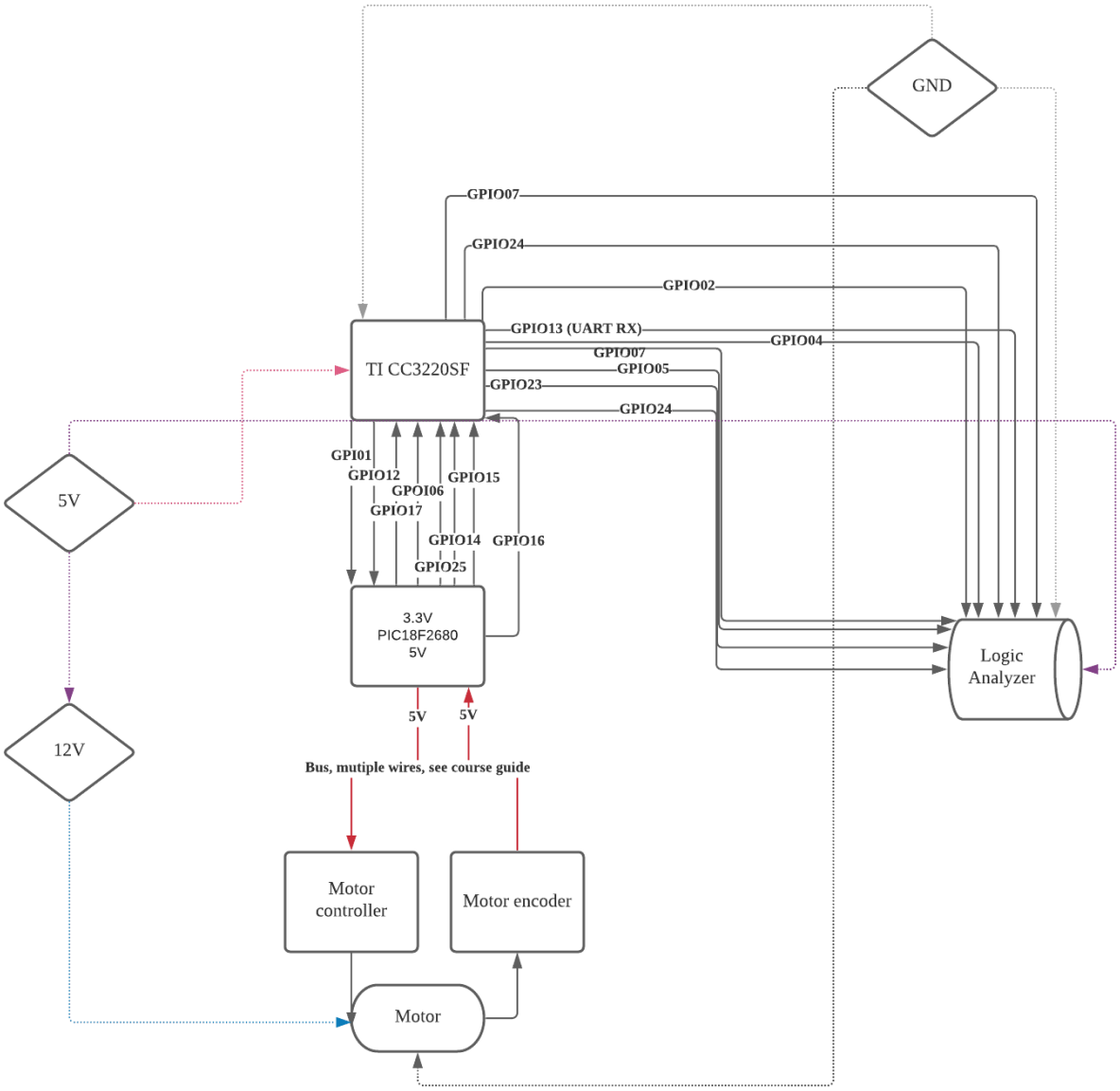
Team 13 Block Diagram

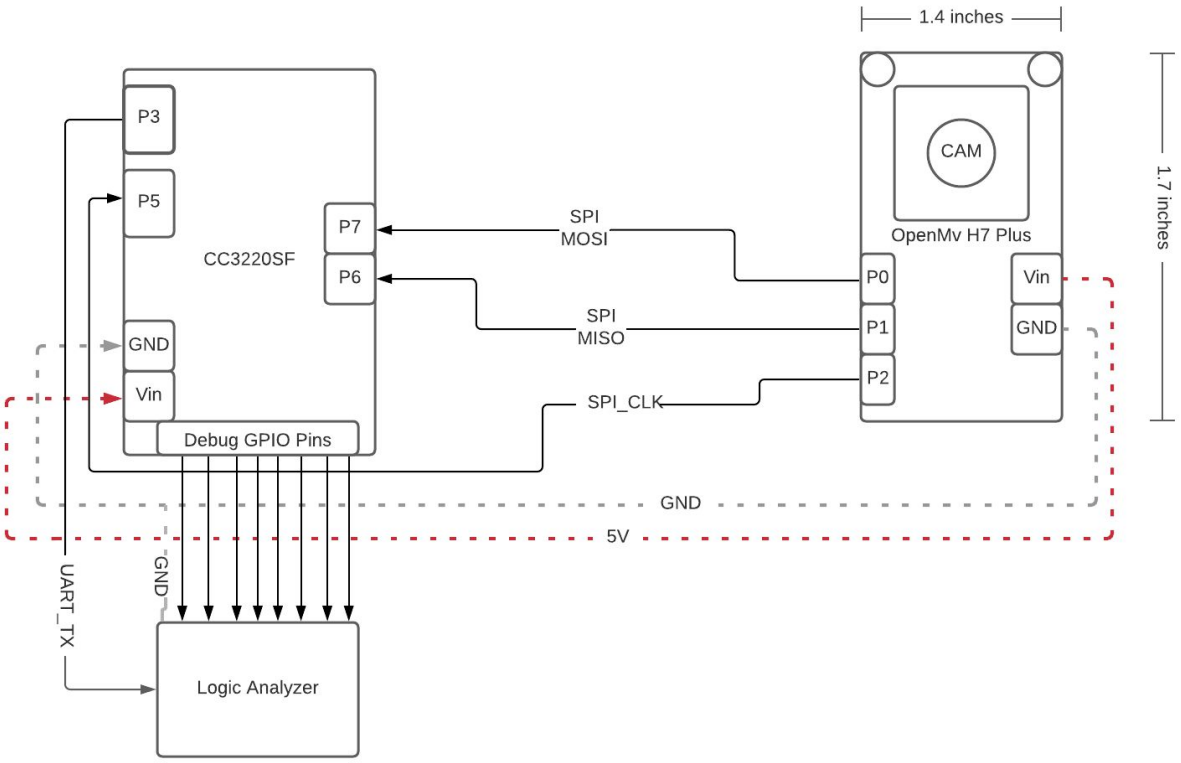
1.Rover (Yifei Liu)



I am now planning to use the Simplified Serial Mode to control the Rover. It uses a UART to control the rover speed and direction. I use the GPIO_12 to send the control signals. And other lines connecting the board and the motor encoder transmit the PID signals telling about the current speed and amount of distance travelled. The board called PIC18F2680 is the voltage change board because the voltage output from the board and controller and encoder standard voltage are different, 3.3V and 5V. And the motor uses 12V to power it. So I have 5V supply, 12 V supply, 3.3 V data and a ground voltage power line in the design. In order to use the logical analyzer to help debug the

program. I have also attached 7 debug lines and one ground line for the logic analyzer. The GPIO13 using UART RX is the UART line for debug use. The logic analyzer also uses the ground from the board and USB power supply from outsider PC.
I am doing the Rover part using Rover and Ti board on the rover. So I do not have a physical diagram. The dimensions for the Rover were fixed when I received the Rover. So I do not include a dimension here.

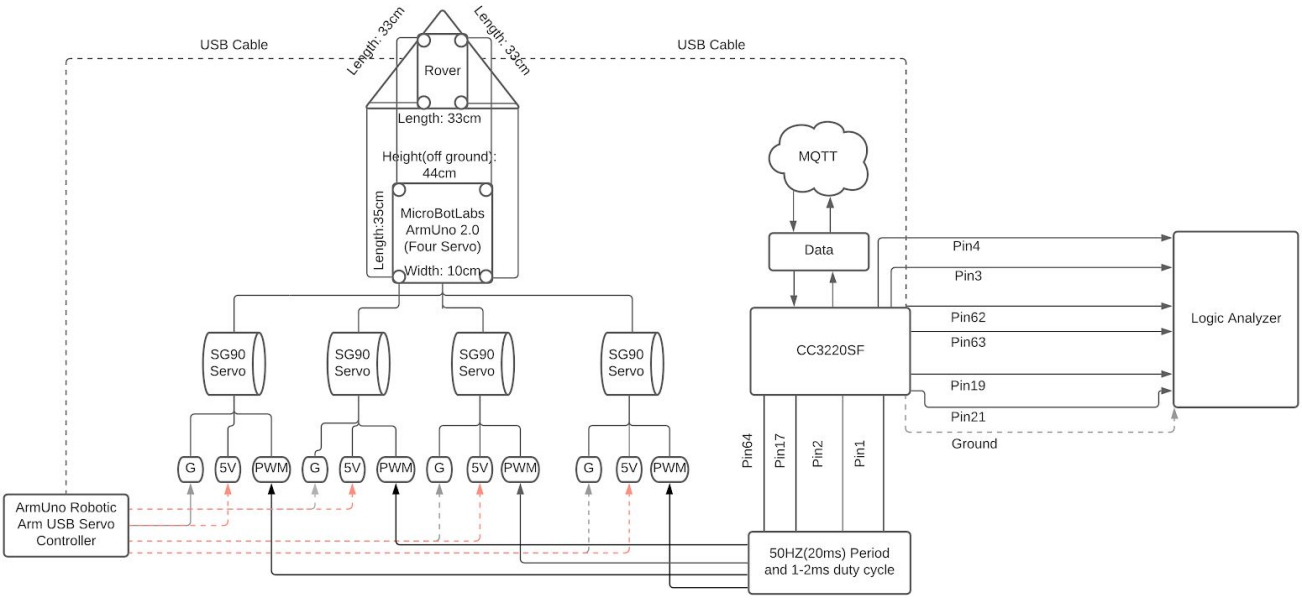
2. Sensor(Zhiyin Liu)



Since our goal of this project contains color detection and distance calculation, we plan on using OpenMv H7 Plus(only version available).
OpenMv has 4 different modes we can use for microcontroller communication, UART, SPI and I2C. Among those ways, SPI has speeds up to 80 Mb/s that allow us to process 1080P images. For SPI protocol, Pin connections between OpenMv H7 and TI CC3220SF are shown above.
Physical diagrams: There are two holes that are used to attach the camera to the Arm. We plan on placing the OpenMV above the two crawls(orthogonal to crawls when they are placed horizontally). The maximum height will not exceed the value for the robotic arm, and the height will be within the height of the robotic arms.

For the wiring, all data lines are hidden behind the camera, goes along the back of the robot arm, which will not cause wires blocking the view of the camera.

3. Robotic Arm (Zhaomeng)



For the robotic arm part, I plan to use MircoBotlabs ArmUno 2.0. It has 35 cm length and 10 cm width, which will be mounted on the rectangular plate of Rover by four screws, so the off-ground height will be 44 cm. This arm needs to be activated by four SG90 servos. Each servo has three jumpers, one for Ground(G), one for 5 Volts, and another one for signal(PWM). However, there is only one 5 volts pin on the CC3220SF, so I plan to use an Arduino board called *ArmUno, MeArm Robotic Arm USB Servo Controller* to give the 5 volts power supplies and Ground pins for these four motors. For CC3220SF, I will generate a 50HZ period PWM that has 1-2ms duty cycles. For 1ms pulse, the motor is all the way to the left; for 1.5 ms pulse,the motor is middle; for 2ms pulse, the motor is all the way to the right. The pins for PWM on CC3220SF are Pin1, Pin2, Pin17, and Pin 64. Both Arduino and MCU should be connected by the USB cable with Rover. The CC3220SF will receive data and send data through MQTT. The Logic Analyzer will use another group of four pins for debug. Pin3 and Pin4 are used for UART.