Team 13

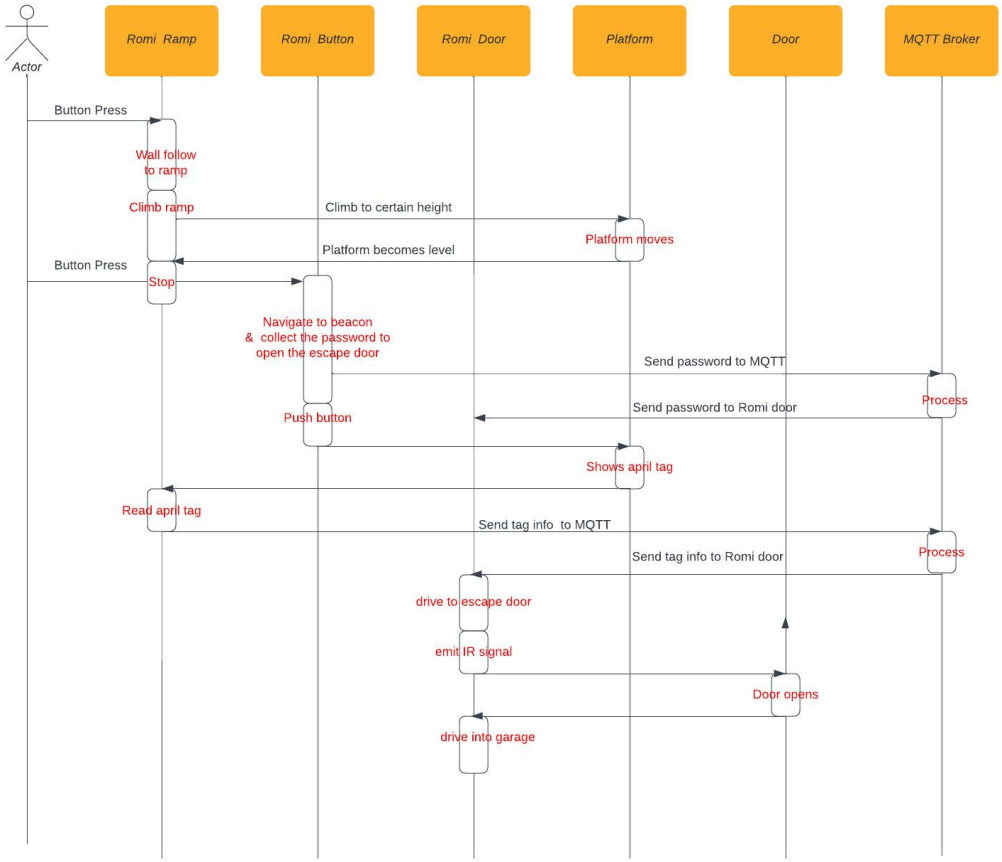
Cole Welcher

Puen Xu

Kevin McCrudden

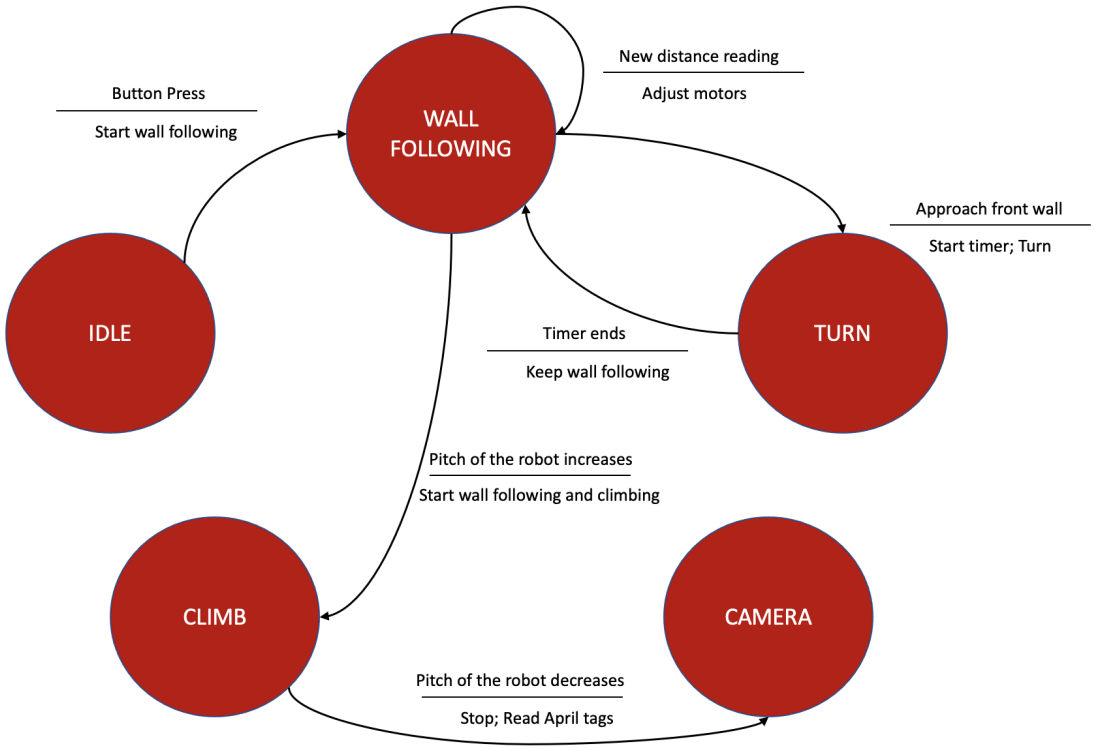
Team 13 RBE 2002 Final Report

Sequence Diagram:



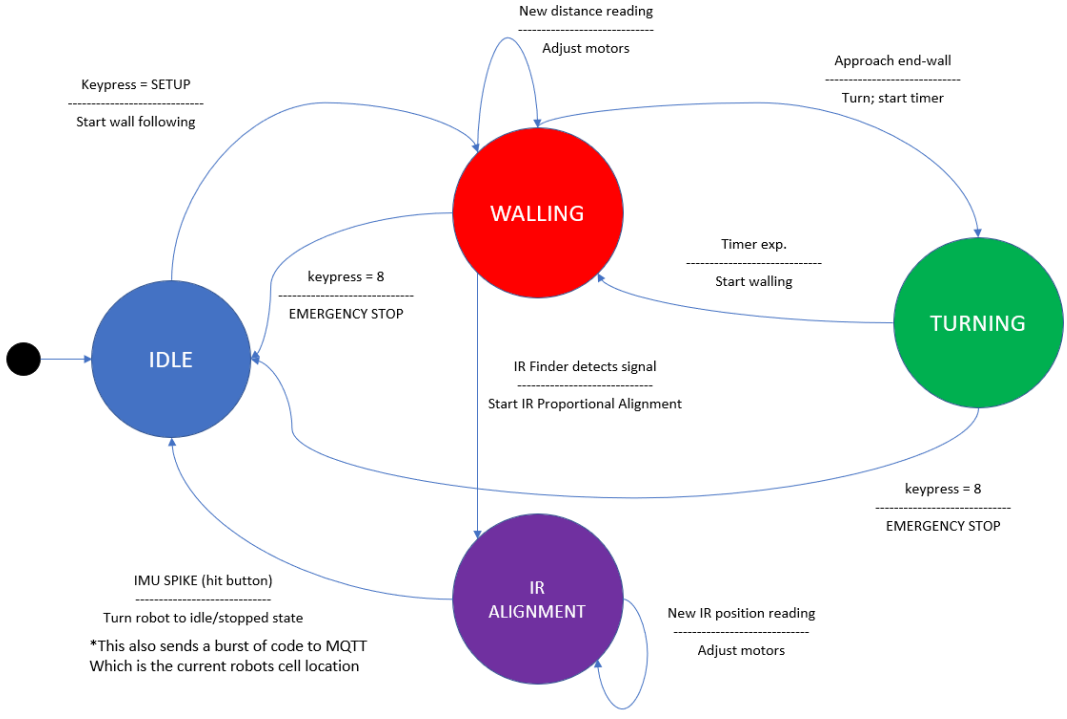
State Machines:

RobotA(Ramp Robot):

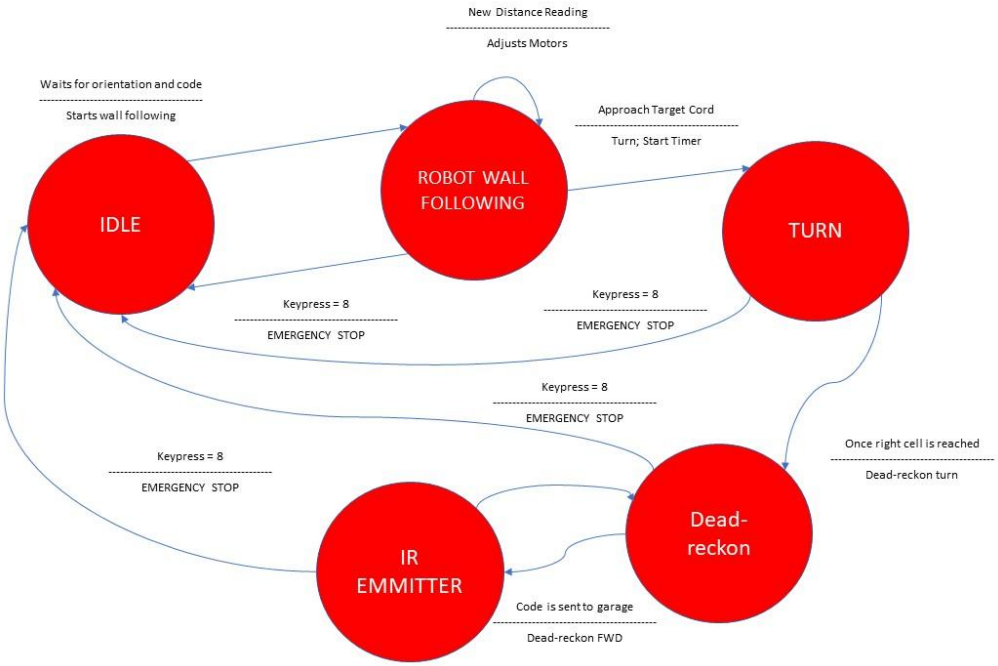


RobotB(IR/Button):

Robot B:

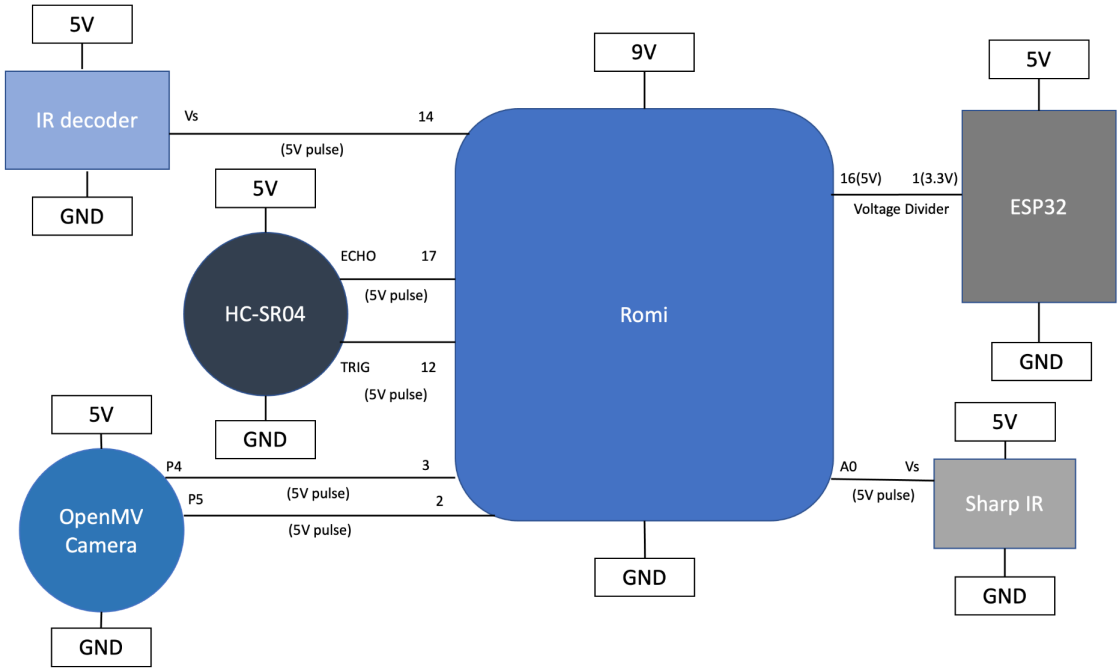


RobotC(EscapeDoor):



Interface Diagrams:

RobotA(Ramp Robot):



OpenMV Camera:

The camera is powered by 5V on the romi to Vin on the camera and shares the same ground with romi. To share April tag information, the camera communicates with the robot via I2C. To be more detailed, P4 (SCL) on the camera is connected to Pin 3 (SCL) on the robot to synchronize; meanwhile, P5 (SDA) on camera is connected to Pin 2 (SDA) on the romi to transmit data. In addition, the reason we choose I2C to interface with the camera is in case that we want to use UART to connect to the MQTT broker to communicate with other robots.

HC-SR04 Rangefinder:

The HC-SR04 rangefinder is powered by 5V and connects to ground on the romi. The TRIG pin on the rangefinder is connected to Pin 12 on the robot, which is an interrupt pin, to be able to receive the trigger signal from the robot. The ECHO pin is connected to an ADC pin on the romi, pin 17, to analog read the voltage. Another major reason that we

choose pin 12 and pin 17 is to make sure all three robots in our team use the same pin configuration so that we can use code from our rangefinder library without making additional changes.

Sharp IR sensor:

The IR distance sensor is powered by 5V from the romi and shares ground with the robot. Its output pin is connected to pin A0 on the romi, which is an analog pin, so that we can read the voltage from the output from the IR sensor to get the distance it is measuring,

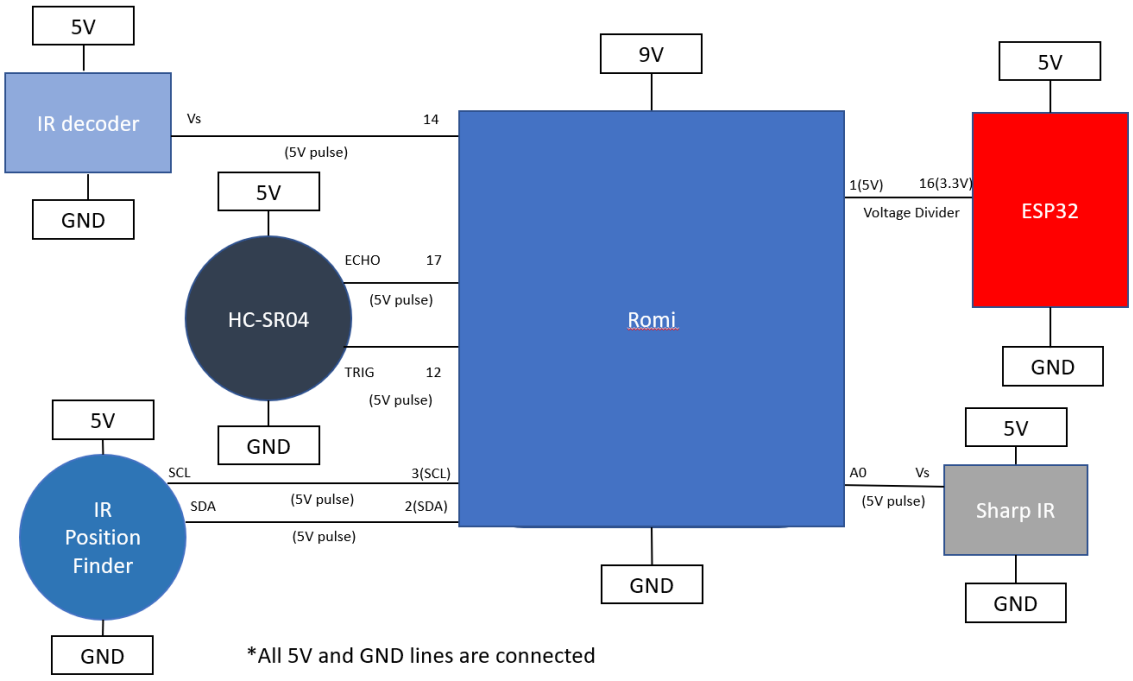
IR decoder:

The IR decoder is powered by 5V and connected to ground. Its Vs pin, the signal pin, is plugged into pin 14 on the robot, which is an interrupt pin to receive IR signal from the remote control.

ESP32:

The ESP32 is powered by 5V from the romi. It is used as a bridge to communicate with MQTT broker via Wi-Fi. To receive the information and send it to the broker, it communicates with the robot via UART. To be specific, RX2 (pin 16) on the ESP32 chip is connected to Pin 1(TX) on the romi with a voltage divider in between. This is because the RX2 pin on ESP32 receives 3.3V pulse and the romi pin 1 outputs a 5V pulse and we want to prevent the 5V pulse from frying the ESP32 chip. Note here, the TX2 from the ESP32 is not connected to the RX on the romi because this specific robot does not necessarily need to receive information from MQTT broker or other robots.

RobotB(IR/Button):



The IR decoder is in port 14, because it requires an interrupt pin so that when we send a signal to it, it will immediately read it via an interrupt. Also plugged into ground and 5V.

The Sharp IR proximity sensor is plugged into ground, 5V, and AO pin, The AO pin is an analog pin which is set to input. Which allows us to see the voltage from the sensor and using an ADC conversion to get a distance based off of the sensor unit/lsb.

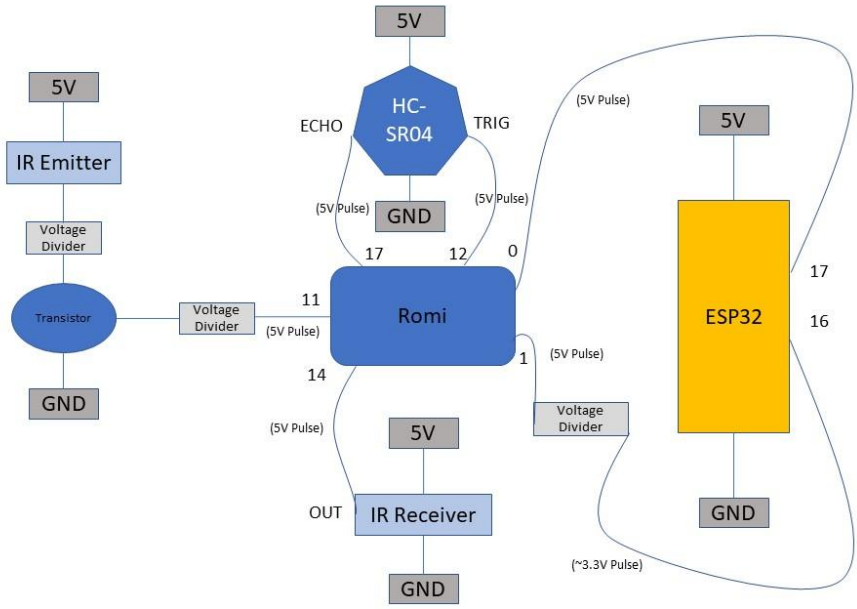
The ESP32 is plunged in on port 1 of the Romi which is a RX pin to receive UART signals, and the ESP32 is plugged in at port 16 which is the TX. They are plugged in via UART because its how the ESP32 communicates with MQTT, if it was I2C we would need to know what we are getting (which we do not). So it must be UART.

The HC-SR04 is plugged in port 12 which is an interrupt pin, so we can trigger the sensor via interrupt. The echo pin is plugged into port 17 which is just a regular

analog pin set to input. We use an ADC conversion and the sensor distance/lsb to find our current distance.

The IR position finder is plugged into 5V, GND, and port 2 and 3. Port 2 is an SDA pin and port 3 is an SCL pin which the IR position finder require so it limits what ports we plug it into. The IR position finder requires I2C communication, SCL for the clock and SDA for the serial data.

RobotC(EscapeDoor):



The ESP32 as seen on the right side of the diagram is used to communicate with the MQTT broker and the Romi. Using pins 16 (RX) and 17(TX) on the ESP32 and pins 0 (RX) and 1 (TX) the Romi can communicate with the ESP32. This allows data to be sent to the MQTT broker and subsequently to other robots. One thing to note on the interface diagram is the use of a voltage divider between Romi pin 1 and the ESP32 pin

16. This is because the pin 1 pulses 5V and pin 16 can only function on ~3.3V, otherwise damage to the ESP32 might occur.

The IR Receiver is yet one of the other crucial components used. It has three pins on the component. One is Vin which is 5V supplied by the Romi, GND which is grounded out on the Romi as well, and lastly the Output which is a 5V pulse back to the Romi. This allows the user to send commands to the Romi and not just turn it on and off. A stop in case of a crash for example.

The HC-SR04 as pictured is used to wall follow on the field. There is four total pins on this sensor, two are for Vin and GND which is 5V and 0V respectively. The other two are the ECHO and TRIG, both are a 5V pulse. Once the ECHO is pulsed the sensor begins timing the time elapsed between pings, this is pulsed on the ECHO pin back to the Romi. The ECHO and TRIG are connected to Romi Pins 17 and 12 respectively.

Lastly there is an IR Emitter, which is used to send a code to the garage door so it opens. The circuit uses Romi pin 11 to send a 5V pulse to the IR Emitter. But it uses a NPN style transistor. Since the Romi can send about 30 mA through that pin it could damage the IR Emitter. Instead we are sending ~20mA by utilizing a voltage divider from Romi pin 11 to pin 2 of the transistor. The emitter is grounded on the Romi, and the gate side has a 5V supply that goes through a voltage divider as well.

Video Link:

Raw Video File:

https://drive.google.com/file/d/19yxbPkdkNZtxFjCZTORNdZJpCurmlZ7a/view?usp=sharing

Voice Over:

https://youtu.be/VdW5u5IIJFw

Code Release:

https://github.com/RBE-2002/final-project-team-13/releases/tag/VF.1