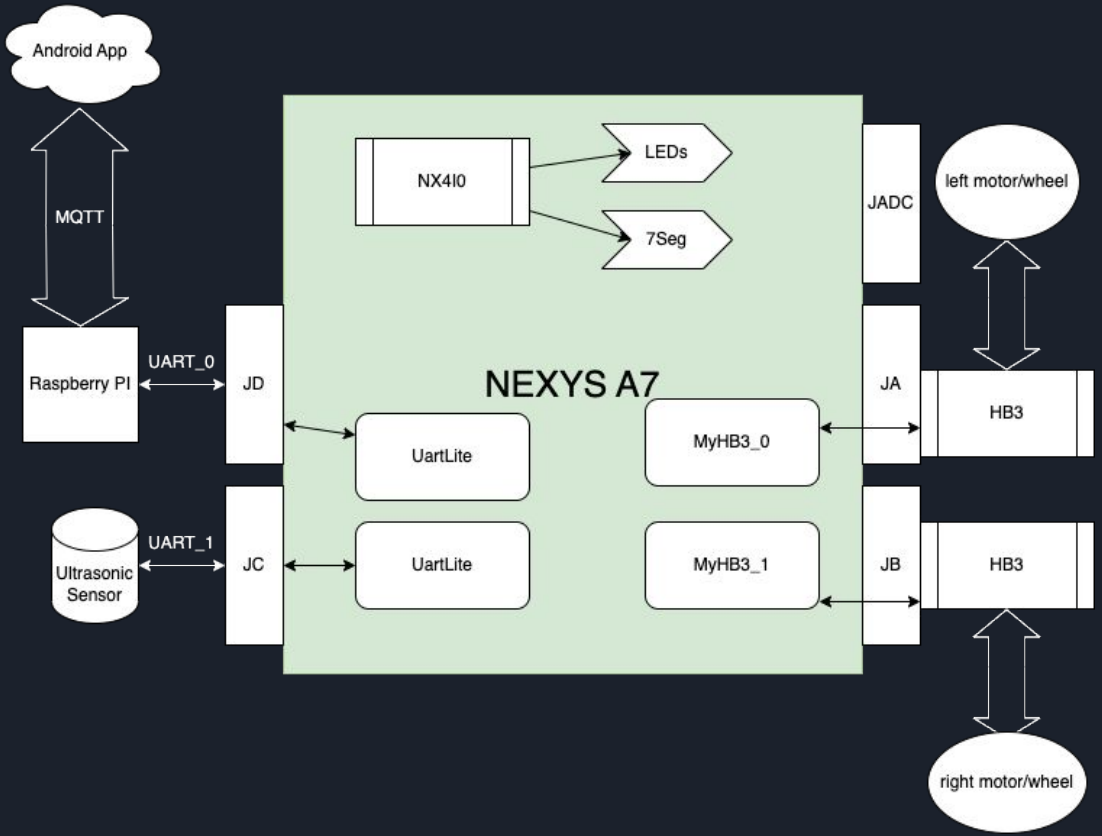
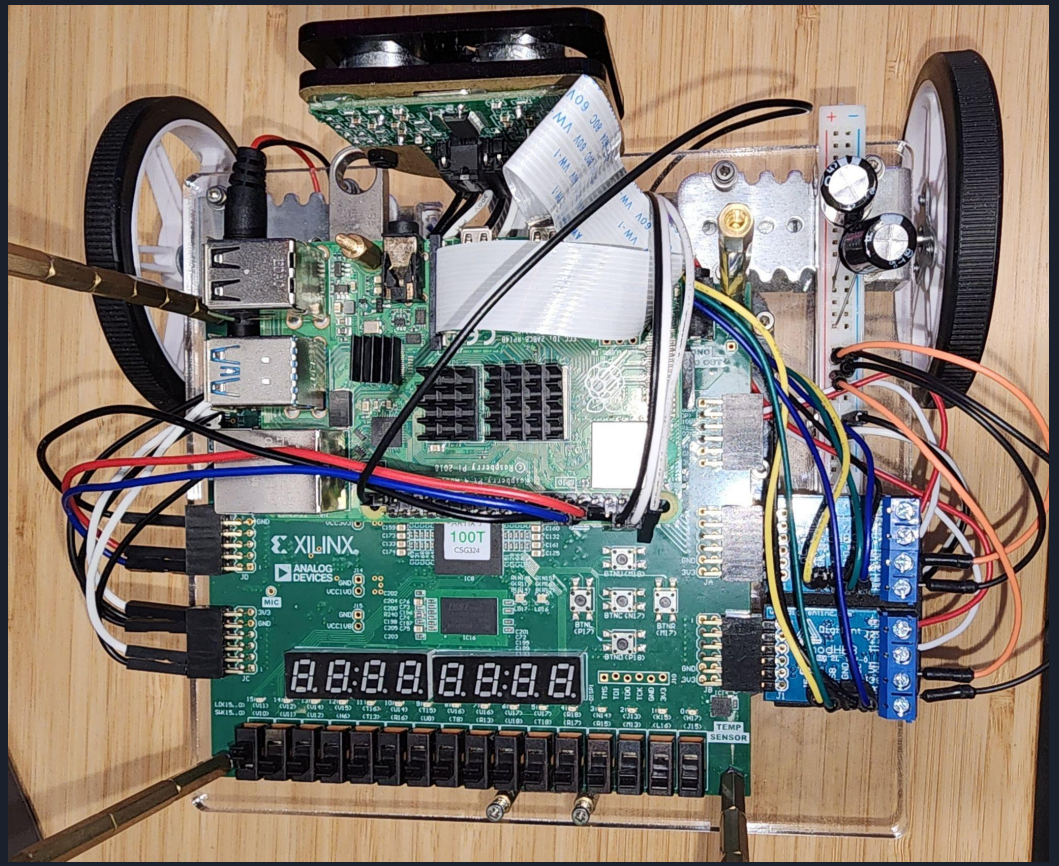
Robot Cars, What Are They Good For? Absolutely Everything!

By Emily Devlin, Noah Page, Drew Seidel, and Stephen Weeks

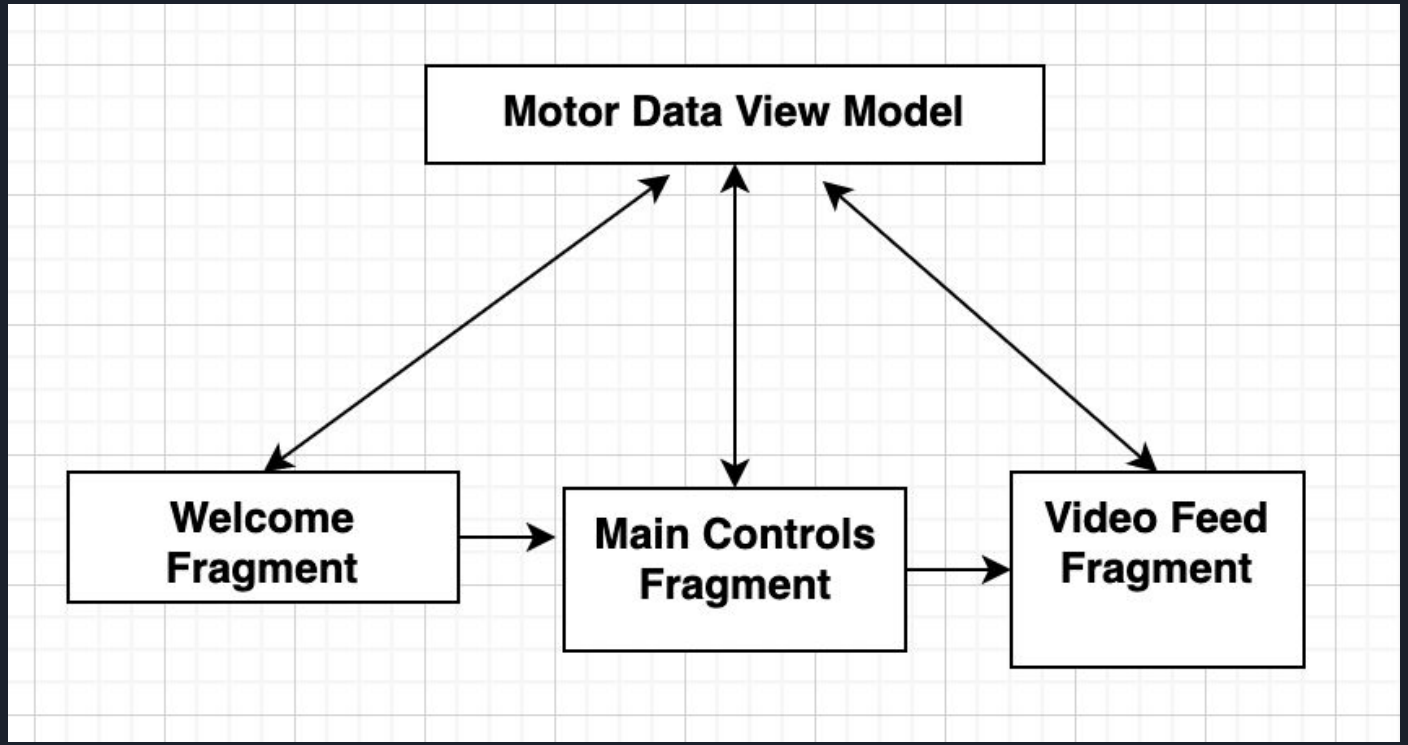
Overall System Hierarchy



Overall System Hierarchy



Android App (Overview)



Android App: Live Data View Model

How to safely declare LiveData variables:

```
private val _motor1_speed = MutableLiveData<Float>()
val motor1_speed: LiveData<Float> = _motor1_speed
```

How to update a LiveData variable in the View Model:

```
fun updateSpeed(RPM: Float, motorNumber: Int) {
    when (motorNumber) {
        1 -> _motor1_speed.value = RPM
        2 -> _motor2_speed.value = RPM
    }
}
```

Shared View Model setup in the Fragment:

```
private var binding: FragmentMainControlsBinding? = null
private val sharedViewModel: MotorDataViewModel by activityViewModels()
```

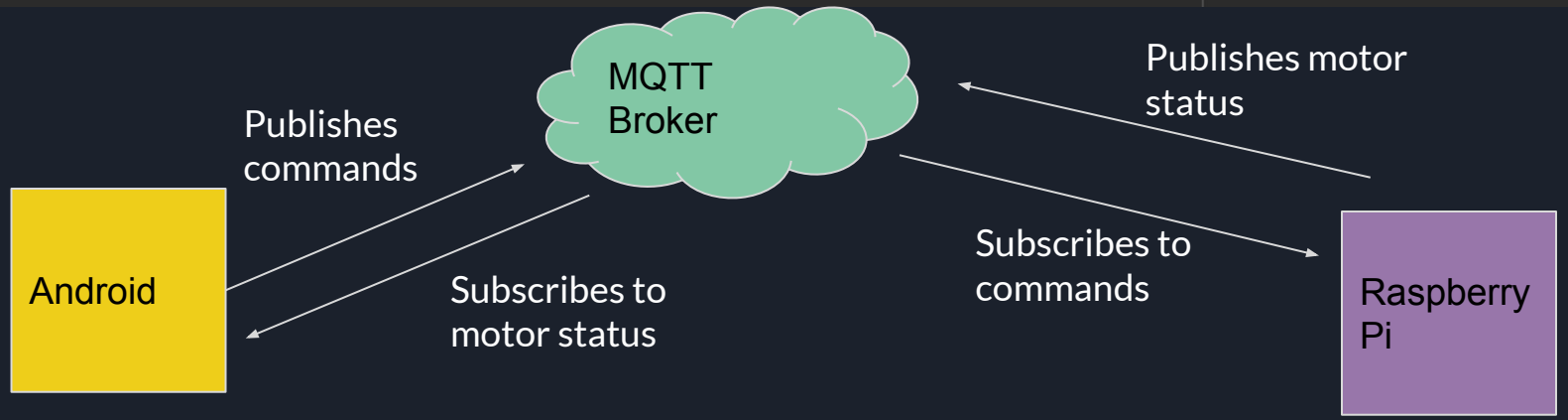
Accessing the variable from the Fragment:

```
sharedViewModel.motor1_speed.observe(viewLifecycleOwner){
    newSpeed -> binding?.speedView?.speedTo(newSpeed)
}
```

Android App (Welcome Fragment/MQTT)

MQTT usage is similar to Project 2, but handled by the View Model and MQTTClient.kt

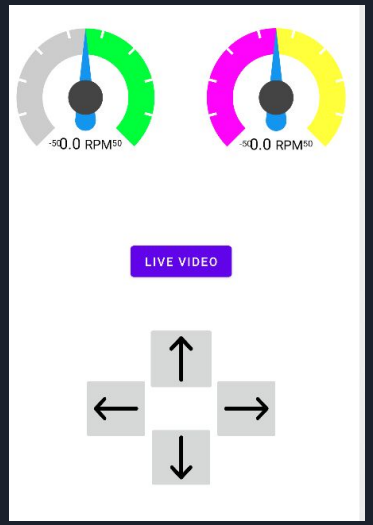
```
// Connect to MQTT using the data view model
sharedViewModel.mqttClientID = MqttClient.generateClientId()
sharedViewModel.mqttClient = MQTTClient(context, sharedViewModel.mqttNetwork.value!!, sharedViewModel.mqttClientID)
sharedViewModel.connectToMQTT()
```



Android App: Main Controls Fragment

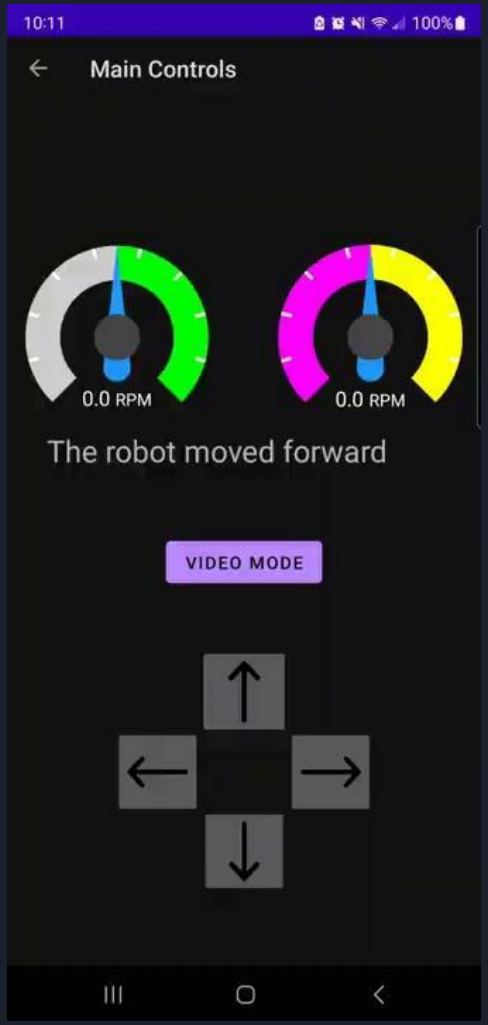
We customized gauges from a library:

https://github.com/anastr/SpeedView




```
binding?.speedView?.apply{ this: SpeedView
    unit = " RPM"
    minSpeed = -50.0F
    maxSpeed = 50.0F
    withTremble = false
    makeSections( numberOfSections: 2, Color.CYAN, Style.BUTT)
    sections[0].color = Color.LTGRAY
    sections[1].color = Color.GREEN
}
```

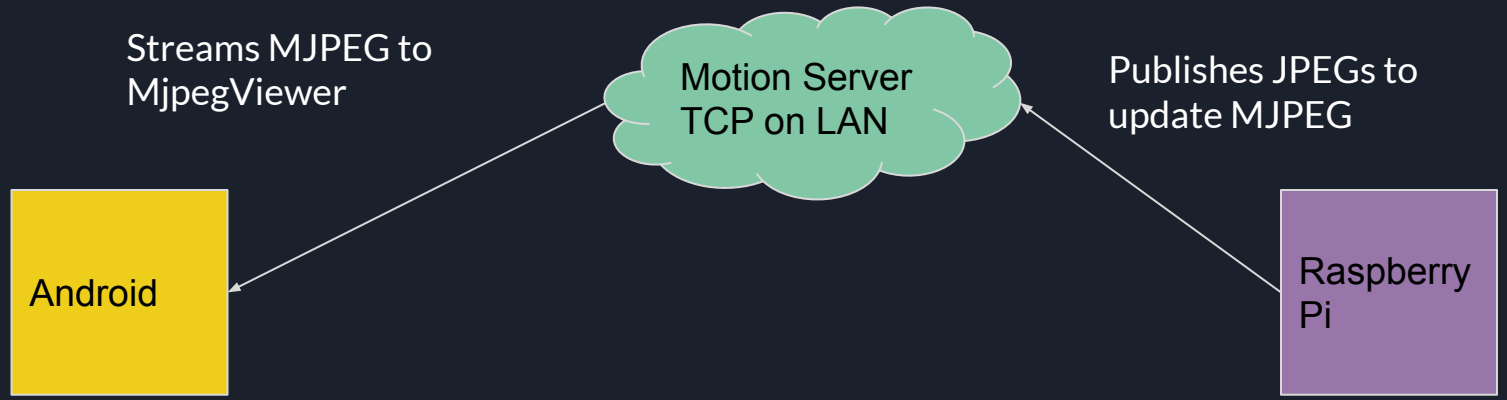
Android App: Main Controls Fragment



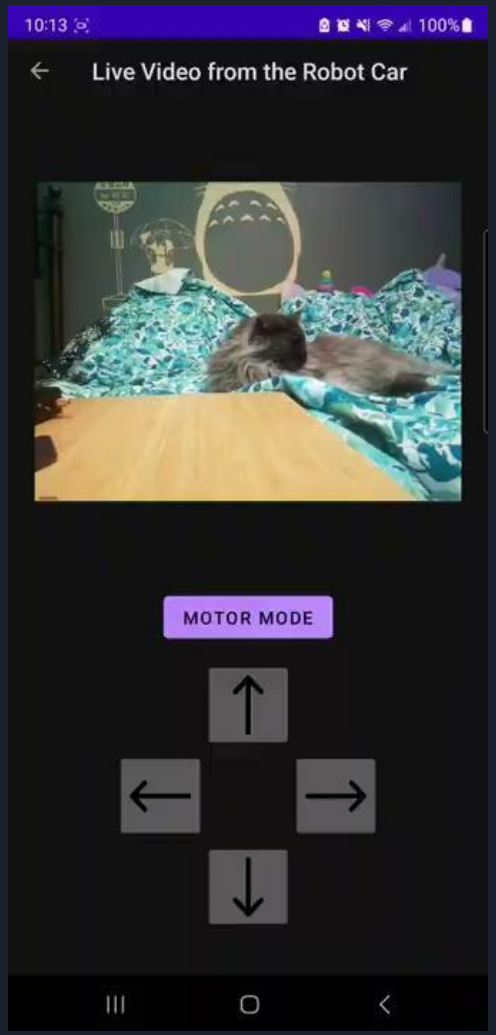
Android App (Video Feed Fragment)

Raspberry Pi runs “sudo libcamerify motion” which sends mjpeg to http://192.168.137.56:8081

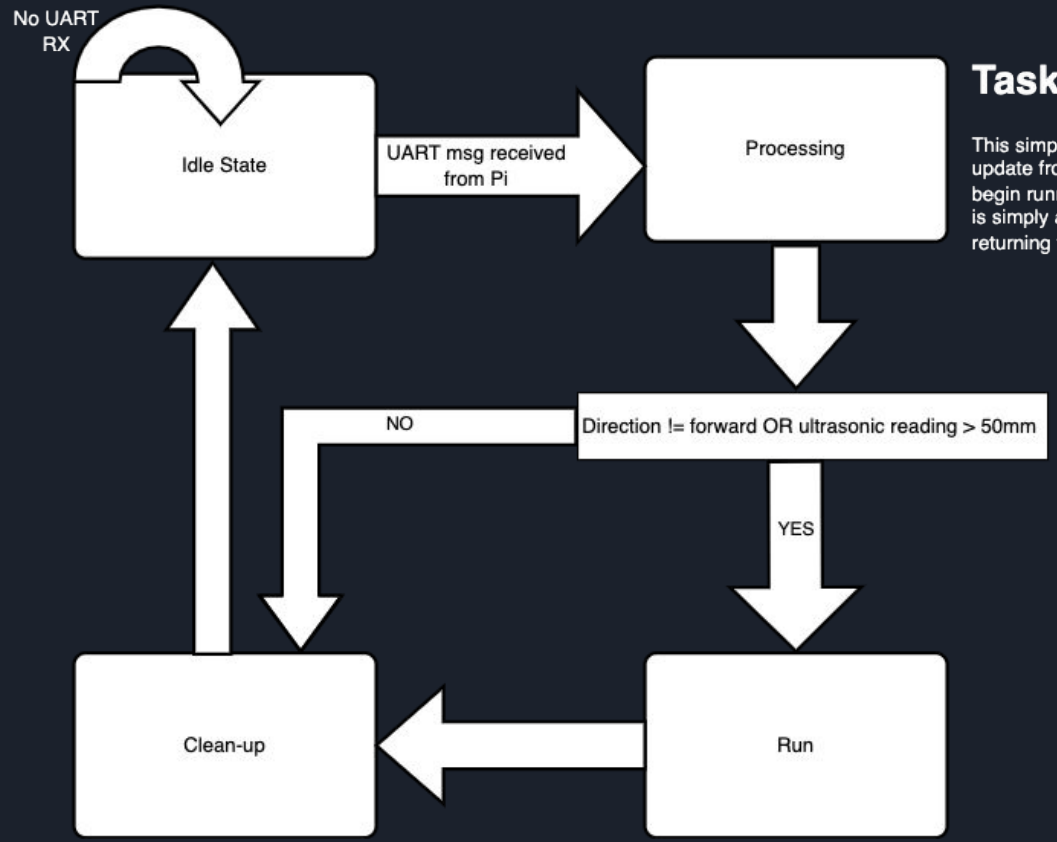
Video Feed Fragment uses MjpegView plugin from https://github.com/perthcpe23/android-mjpeg-view



Android App: Video Feed Fragment



FPGA FSM



Task Scheduler FSM

This simple task scheduler waits for an update from the Raspberry Pi via UART to begin running, for right now the running state is simply a timed based input before returning to the idle state.

FPGA FSM Method and Run State

```
/*****************Global State*****************/
task_t run_state_t = idle;
void (*task_scheduler[4])() = {idle_state, processing_state, run_state, end_state};
```

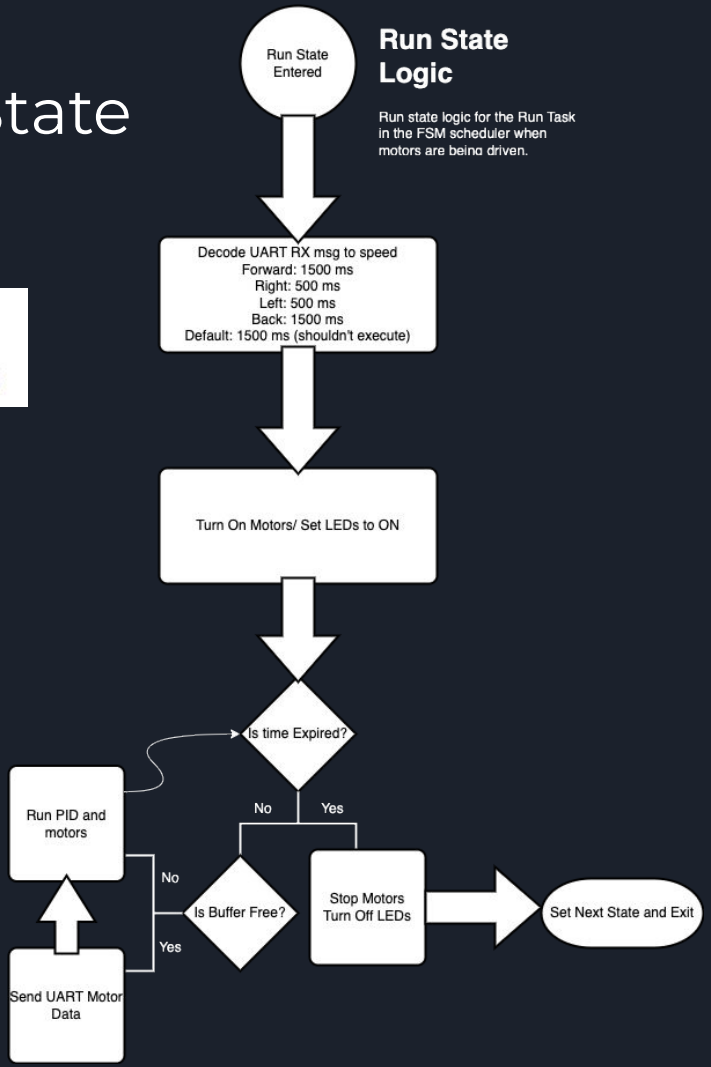
Function pointers for each FSM state

```
while(1)
{
    task_scheduler[run_state_t]();
}
```

Main infinite forever loop

```
typedef enum
{
    idle,
    processing,
    run,
    end,
} task_t;
```

Enum for indexing into the task scheduler with descriptive names

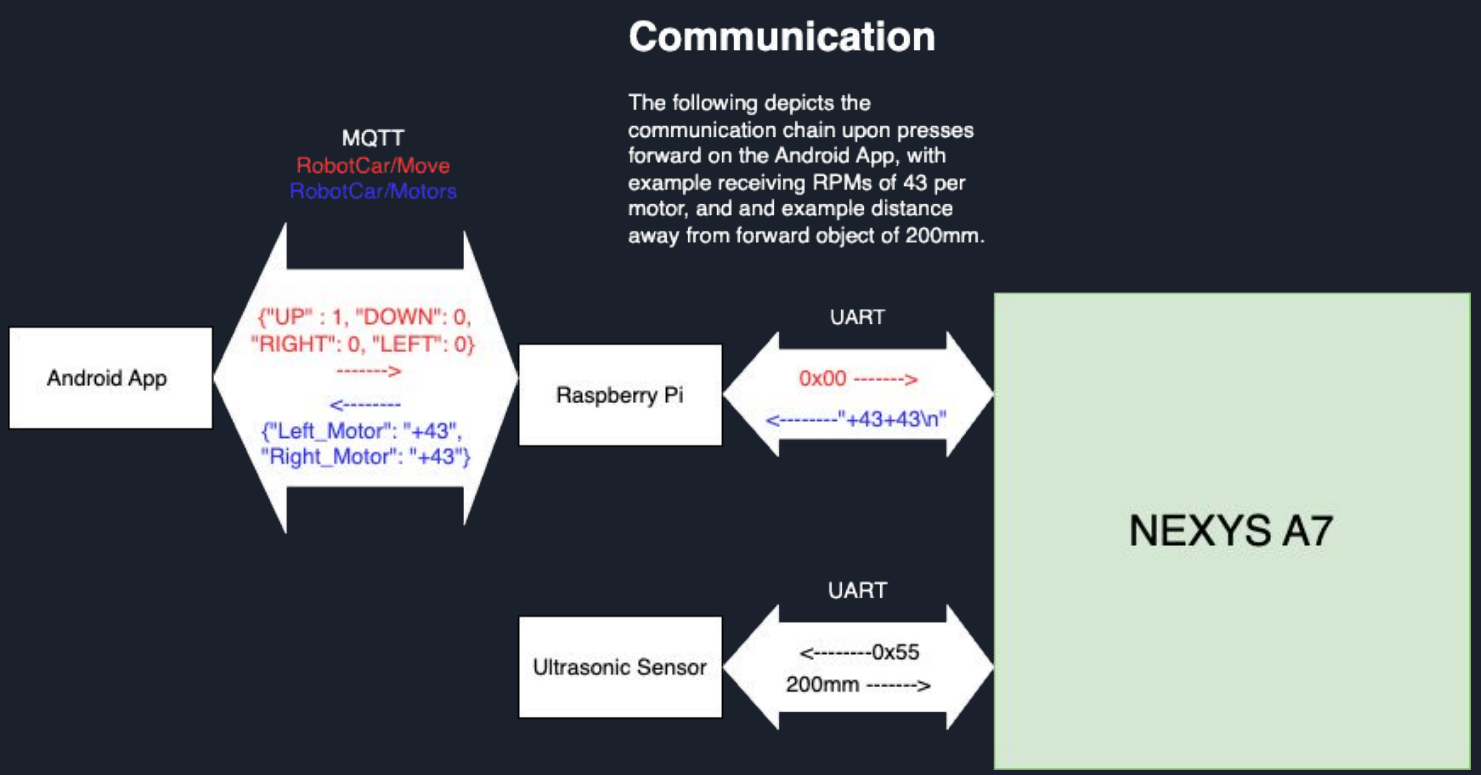


Communication

Five major intercommunication systems were designed for this system:

- MQTT Topic RobotCar/Move
  - Android App publisher, Raspberry Pi subscriber (single JSON object)
  - Used to control the motors
  - JSON object example message: {“UP”:1, “DOWN”:0, “RIGHT”:0, “LEFT”:0}
- MQTT Topic RobotCar/Motors
  - Raspberry Pi publisher, Android App subscriber (single JSON object)
  - Used to display RPM information from the motors to the Android App display
  - JSON object example message: {“Left_Motor”, “-43”, “Right_Motor”, “+43”}
- UART Bus Between FPGA and Raspberry Pi
  - Raspberry Pi Tx to FPGA Rx decodes received RobotCar/Motors message and translates that to single byte direction command
  - FPGA Tx to Raspberry Pi Rx sends 5 byte buffer with RPM and direction information for each motor. This is translated into the single JSON object to be published to RobotCar/Motors by the Pi
- UART Bus Between FPGA and Raspberry Ultrasonic sensor
  - FPGA Tx to Sensor Tx - single byte (0x55) is sent to receive two-byte millimeters value
  - Sensor Rx to FPGA Tx - two byte buffer is received and 16-bit value for millimeter evaluated

Communication



Bill of Materials:
Retail Cost (not including shipping)

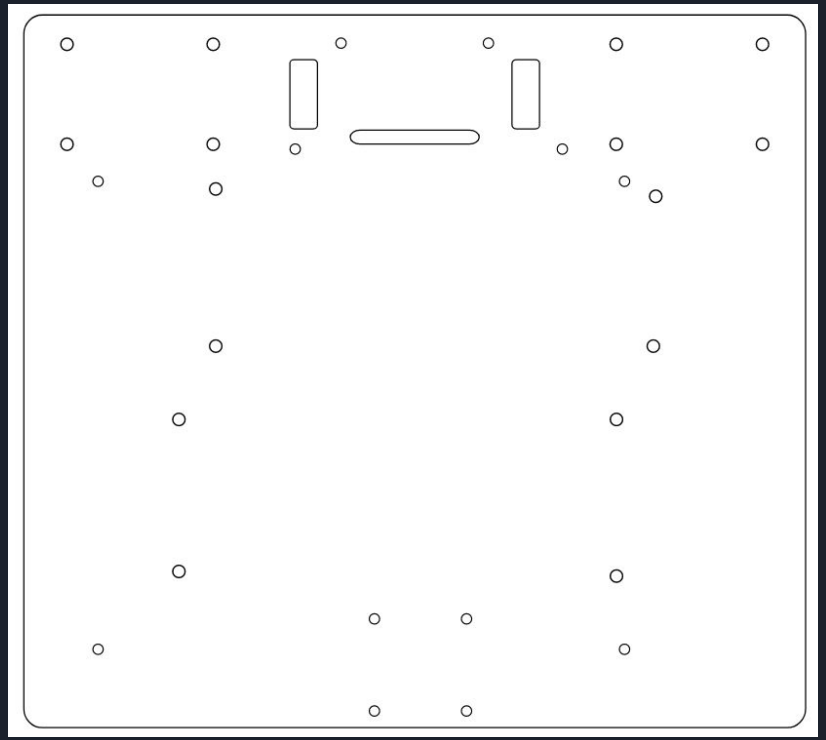
Component	Vendor	Retail Cost	Notes
Nexys A7 FPGA	Digilent	$349	https://digilent.com/shop/nexys-a7-fpga-trainer-board-recommended-for-ece-curriculum/
Raspberry Pi 4 Model B 4GB	Cytron	$75	https://thepihut.com/products/raspberry-pi-4-model-b?variant=20064052740158&src=raspberrypi
Pi Camera Module 2	Cytron	$31.25	https://www.cytron.io/p-raspberry-pi-8mp-camera-module-v2
Metal DC Geared Motor w/Encoder 6V 100RPM (x2)	DFROBOT	$39.80	https://www.dfrobot.com/product-1618.html
2-Way 18650 Battery Holder (x2)	DFROBOT	$19.80	https://www.dfrobot.com/product-2578.html
25D mm Metal Gearmotor Bracket Pair	Pololu	$7.95	https://www.pololu.com/product/2676
Multihub Wheel Pair 80x10mm	Pololu	$9.95	https://www.pololu.com/product/3691
Tamiya 70144 Ball Caster Kit	Pololu	$7.00	https://www.pololu.com/product/66
US-100 Ultrasonic Distance Sensor 3.3V	Adafruit	$6.95	https://www.adafruit.com/product/4019
Epoch 30P 18650 Battery (x4)	18650batterystore	$23.96	https://www.18650batterystore.com/products/epoch-30p-18650
Pmod HB3 (x2)	Digilent	$19.98	https://digilent.com/shop/pmod-hb3-h-bridge-driver-with-feedback-inputs/
0.25" Acrylic Sheet	EPL	$15.00	
M2.5 Spacer/Standoff Assorted Kit	Amazon	$11.99	https://www.amazon.com/HanTof-Raspberry-Installation-Standoff-Accessories/dp/B07KM5B3PT/
M2.5 Nuts/Bolts Assorted Kit	Amazon	$12.99	https://www.amazon.com/dp/B082XPZV1V?psc=1&ref=ppx_yo2ov_dt_b_product_details
Various wires and Capacitors	-	-	-
	Total Cost	$631	

Robot Design

The components picked were largely based on what we already had for ECE 544 Project 2

After the components arrived, prototype shapes for the chassis were made and once finalized an SVG file was created so that it could be laser cut

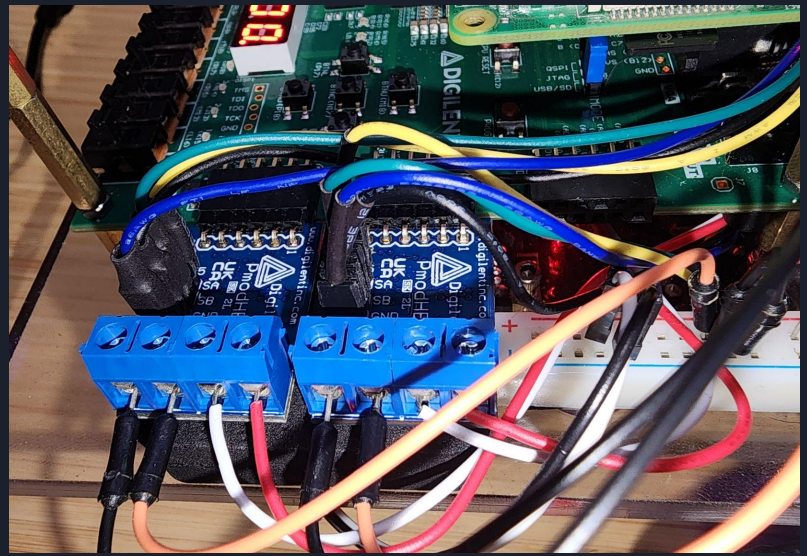
The 3rd iteration seen here has mounting holes for all the components and larger gaps for the pass through of wires

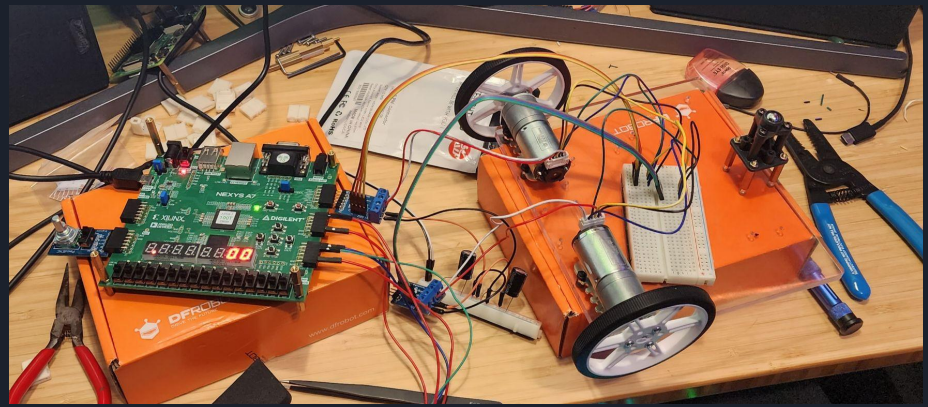


Robot Design

Our goal was to limit wire lengths and reliance on breadboards

Tidying up the wires and some of the more difficult solders





Committed Goals

Goal	Project Area	Goal Type	Success?
Two-way UART communication between FPGA and Raspberry Pi	FPGA	Committed	Yes
Two-way MQTT communication between RPi and Android	Android	Committed	Yes
Fragment Navigation	Android	Committed	Yes
4 arrow buttons to move robot	Android	Committed	Yes
Ultrasonic sensor hardware in FPGA to avoid crashes	FPGA	Stretch	Yes
Raspberry Pi camera live streaming video	RPi/Android	Stretch	Yes
Display screen with info about motors	Android	Stretch	Yes