FPR

UMass Amherst
College of Engineering

TEAM 8

AUTOMATIC PET CAGE

UMass Amherst
College of Engineering

PROBLEM STATEMENT

Adding food and water for a pet is always a trouble for their owners. Forgetting to add food and water may sicken pets, or even causes the death of pets. So here we designed a smart pet cage that can add necessities for pets.

UMass Amherst
College of Engineering

system specifications

1.Add Food.

a. Add two kinds of food for guinea pig

b. When food remains low, send message to user

2.Add Water.

a. Keep replenishing water in the water bottle for guinea pig

b. When water in water tank is little, send message to user

c. monitor the volume of water that guinea pig drinks

3.Remote Status Report.

a. Collect data from sensor and send them to user's phone

b. Send notification to user when the data is out of range

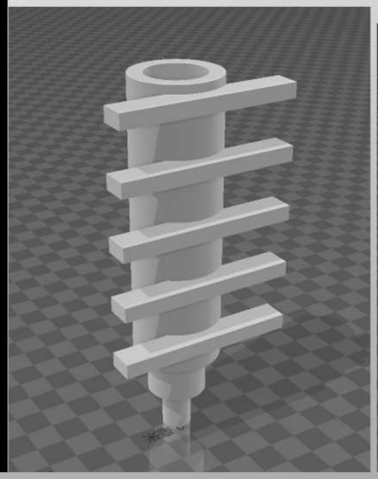
Documentation of Add Hay Machin

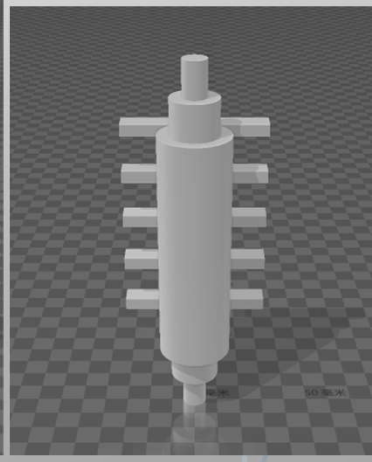
Documentation

20%	Rubric
• Describes **final system**	**(4.0)** Documentation clear and com
• Diagrams & lists (HW/SW)	**(3.0)** Some details are missing or u
• PCB schematic, board layout	**(2.0)** Significant details missing or
• Other relevant documentation	**(1.0)** Documentation is unsatisfacto

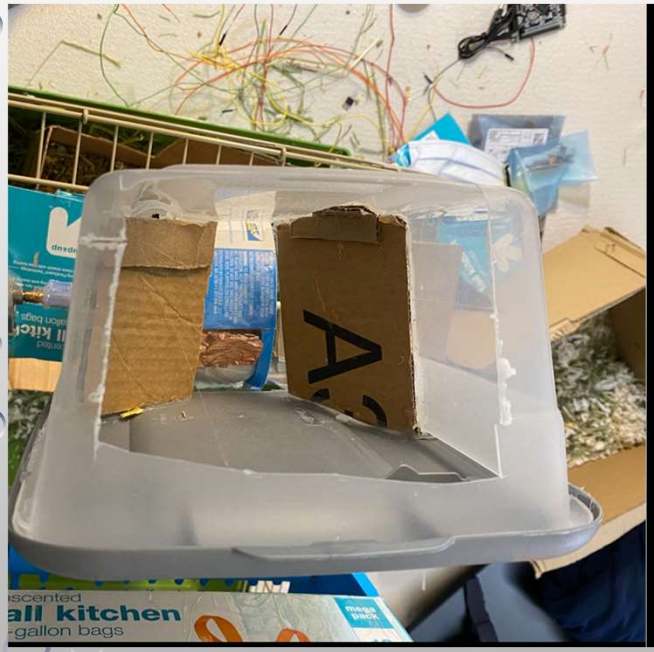
2x 12v high-torque motors

2x 4007 Diodes





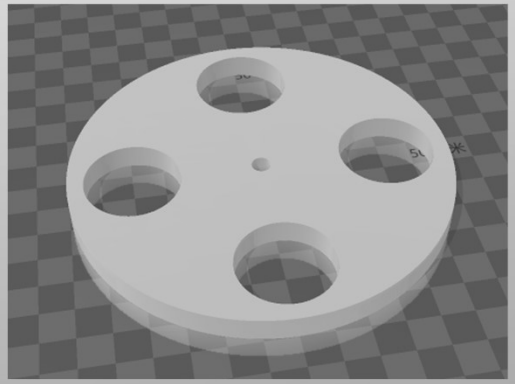
Documentation of Add Food Machine



Component Usage:

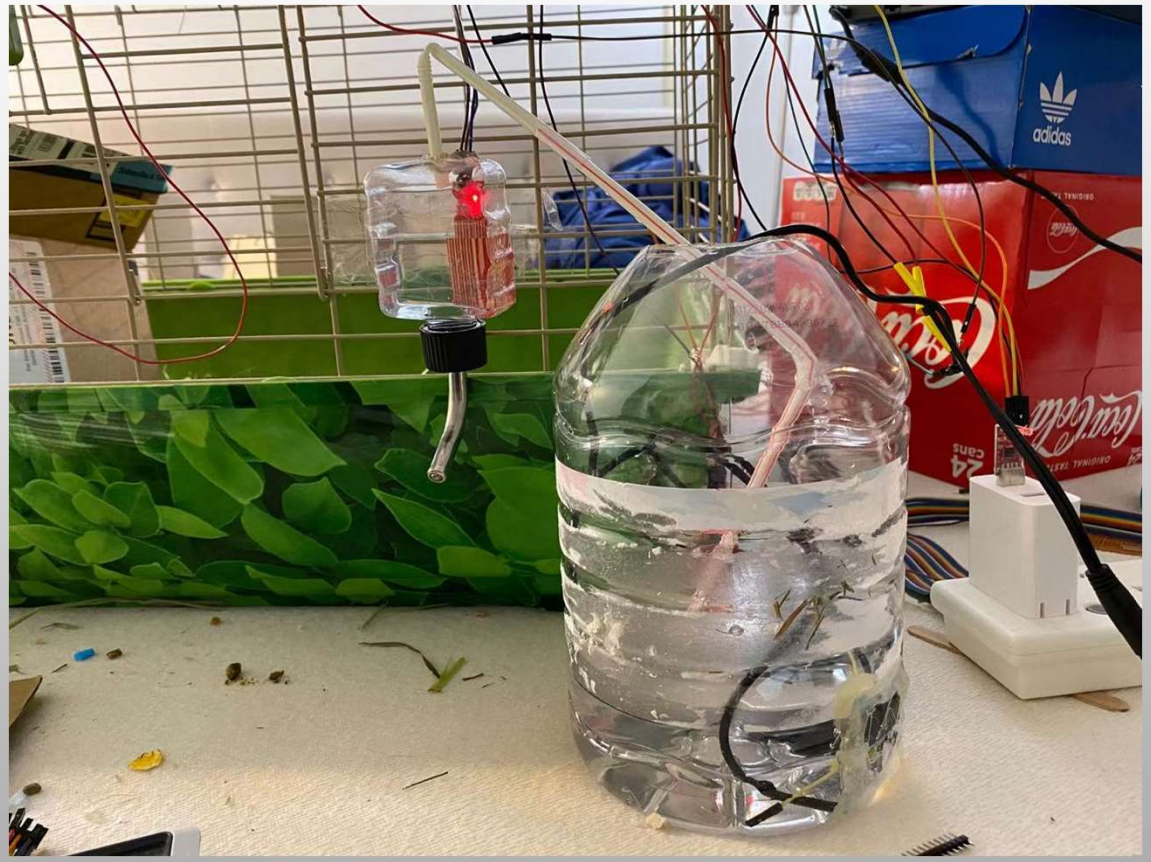
1x 12v high-torque motors

1x 4007 Diode



Documentation of Auto-Add Water Machine

Add water system



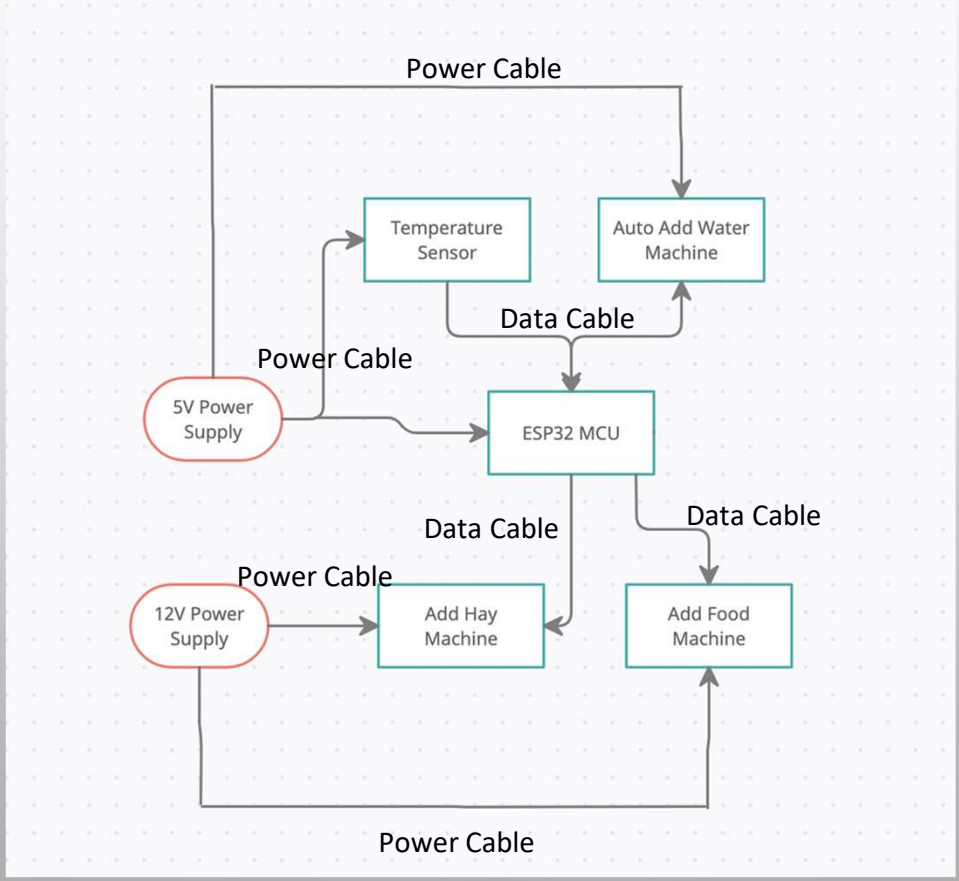
Component Usage:

1x 5v mini pump

1x Water Level Sensor

1x 4007 Diode

Documentation of Hardware Diagram

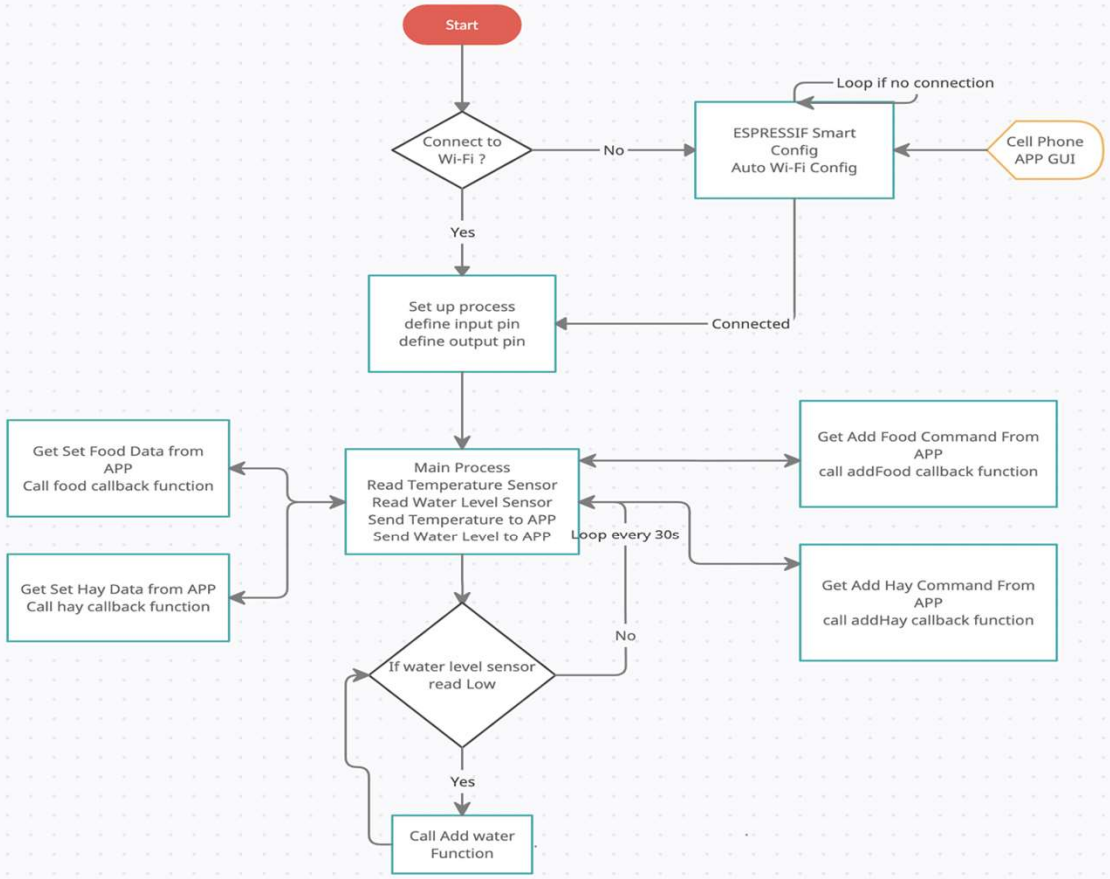


Documentation of Hardware List

UMass Amherst
College of Engineering

Part Name	Model	Quantity	Voltage/Current
High torque motor	General	3	12V / 600mA
Diode	4007	4	
Temperature Sensor	TMP36	1	5V
Water Level Sensor		2	3.3V
ESP32 MCU	ESP32-S2	1	3.3V
NMOS	IRF520		<24V / <5A
mini Pump	USB	1	4.7V / Rate of flow:0.02L/s
AMS1117		3	12V to 5V and 5V to 3.3V
CAP CER 0.1UF	X7R 0805		100V
USB to UART Bridge	28VQFN	---	OUT OF STOCK

Documentation of Software Diagram



Documentation of Software

UMass Amherst
College of Engineering

GitHub Repo
https://github.com/shenttt123/SmartPetCage

Firstly, we use micropython to develop wifi connection. After I successfully configured wifi, I found there are some serious problems when using micropython. It is not suitable to develop a iot device using high-level programming language due to weak performance. Lack of Blinker library support is another problem.

Then we changed to C++, and we could use esp32 official esptouch to establish wifi connection. Users don't need to first connect esp32 ap hotspot, they could establish wifi connection through phone app instead.

Blinker provides customizable button for developers to use. We only need to edit button's text, icon and write callback function for that button.

If we publish this project, other users could add out smart pet cage directly from the app, configure wifi, and start using it immediately. The auth key will be automatically distributed.

Documentation of Software

UMass Amherst
College of Engineering

```
#define Slider_1 "Food1Amount" //food slide1
#define Slider_2 "Food2Amount" //food slide2
BlinkerSlider Slider1(Slider_1);
BlinkerSlider Slider2(Slider_2);
BlinkerNumber temperature("temperature");
BlinkerNumber waterRemain("waterLevel");
BlinkerButton Button1("addFoodBnt1");
BlinkerButton Button2("addFoodBnt2");
```

Define buttons and sliders used in the app

```
void slider1_callback(int32_t value){ // read the value
from user's app
   BLINKER_LOG("get slider value: ", value);
   fA1 = value;


}
void button1_callback(const String & state){// add hay
   BLINKER_LOG("get button state: ", state);
   if (state == "tap") {
   digitalWrite(25, HIGH);
   digitalWrite(26, HIGH);
   Blinker.delay(50*fA1);// action time is set by user's
app silder
   digitalWrite(25, LOW);
   digitalWrite(26, LOW);
 }
}
```

Silder and button callback function

Documentation of Software

UMass Amherst
College of Engineering

```
void heartbeat()
{
   BLINKER_LOG("heartbeat!");
   float tempval = getTemp();
   float waterTankLevel = getTankLevel();
   temperature.print(tempval);
   waterRemain.print(waterTankLevel);
   Slider1.print(fA1);
   Slider2.print(fA2);
}
```

heartbeat function, app send heartbeat to board, board return status

```
float getTemp(){
 float tempR1 = float(analogRead(34));
 float a1 = (tempR1 *0.73 / 700 -0.5)* 100 - 2.5;
 float a = a1;
 int i = 0;
 for(i;i<20;i++){
    tempR1 = float(analogRead(34));
    a1 = (tempR1 *0.73 / 700 -0.5)* 100 - 2.5;
    a += a1;
 }
 return (a*1.8/21+32)+8;
}
```

```
float getTankLevel(){
 float waters = float(analogRead(32));
 float w1 = waters *0.73 / 700;
 float w = w1;
 int ii = 0;
 for(ii;ii<20;ii++){
    waters = float(analogRead(32));
    w1 = waters *0.73 / 700;
    w += w1;
 }
 return w/21*100;
}
```

read and convert temp and water level from sensor

Documentation of Software

UMass Amherst
College of Engineering

```
void dataStorage() {
 float tempval = getTemp();
 float waterTankLevel = getTankLevel();
 Blinker.dataStorage("temperature", tempval);
 Blinker.dataStorage("waterLevel", waterTankLevel);
}
```

Store data to the cloud for future graph use

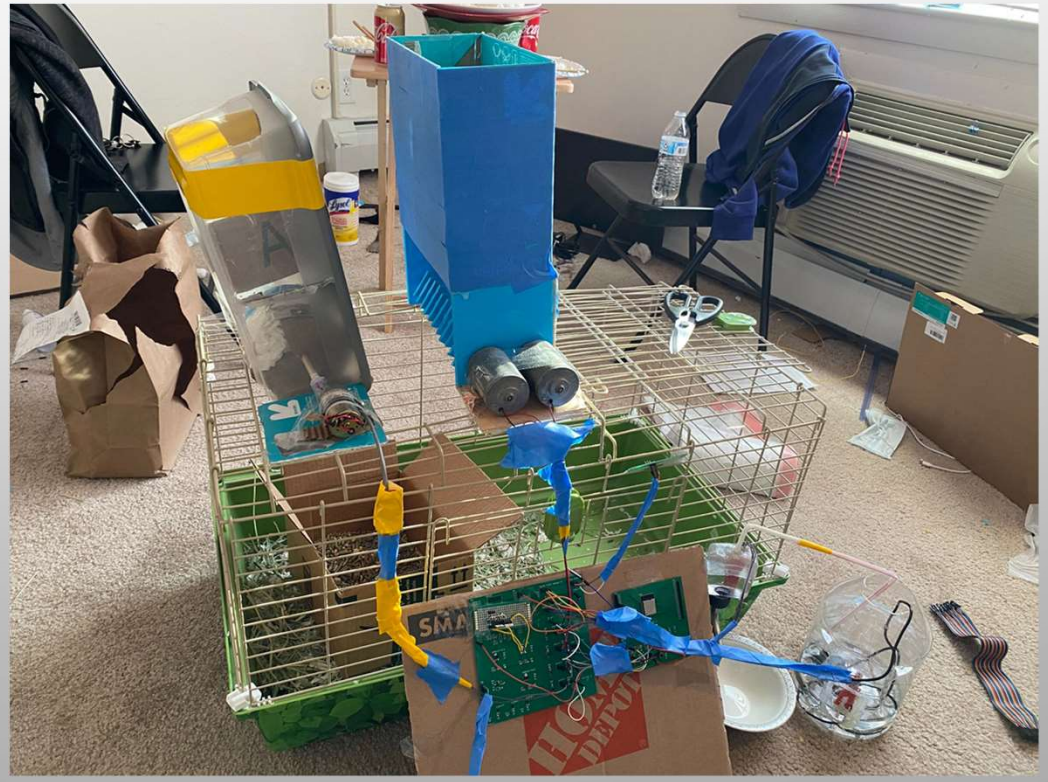
```
if (waterLevel < 1){
    Blinker.push("Pet Notice: Low water Remain!");
}
```

Send notification when waterLevel sensor reading is low

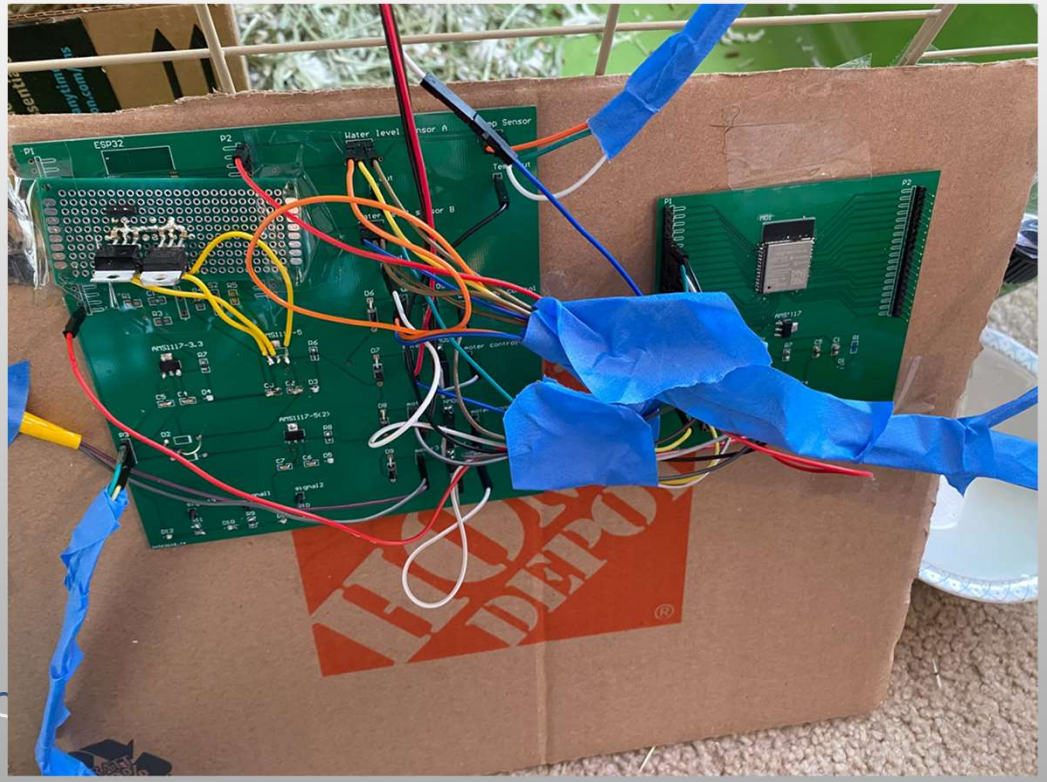
```
void setup()
{
   Serial.begin(115200);
   BLINKER_DEBUG.stream(Serial);
   BLINKER_DEBUG.debugAll();
pinMode(33, OUTPUT);// Pump Pin
pinMode(25, OUTPUT);// Add Hay1
PinMode(26, OUTPUT);// Add Hay2
pinMode(27, OUTPUT);// Add Food
pinMode(34, INPUT);// Temperature pin
pinMode(32, INPUT); // WatertTankSensor PIN
pinMode(35, INPUT); // Smallwatertank Sensor PIN


   Blinker.begin(auth, type);
   Blinker.setTimezone(-4.0);// set TimeZone as EST
   Blinker.attachHeartbeat(heartbeat);
   Blinker.attachDataStorage(dataStorage, 120 , 1); //
120s a cycle to transmit data to the cloud
   Slider1.attach(slider1_callback);
   Slider2.attach(slider2_callback);
   Button1.attach(button1_callback);
   Button2.attach(button2_callback);


}
```

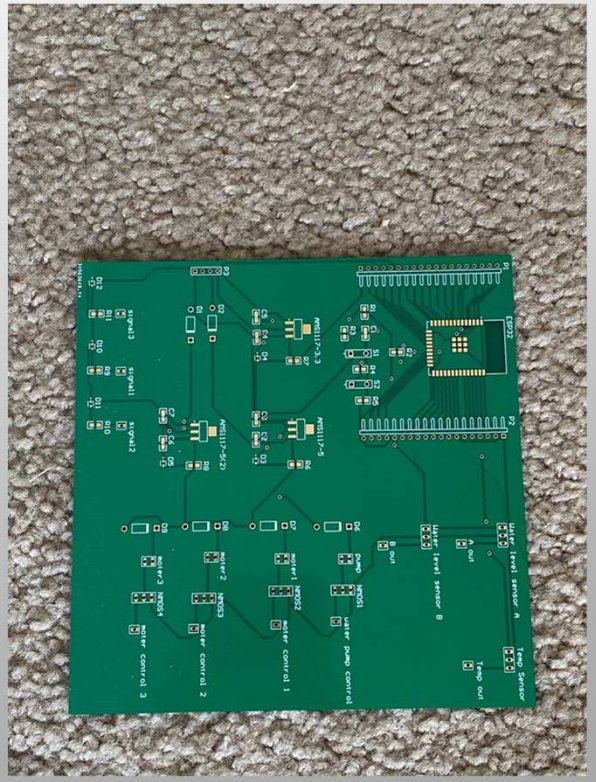
Initialize blinker, set pin and register heartbeat storage and callback function.

Integrated System (Hardware Overview)

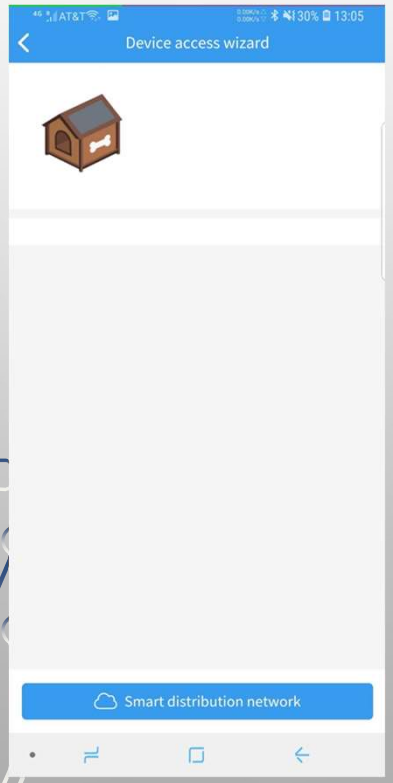


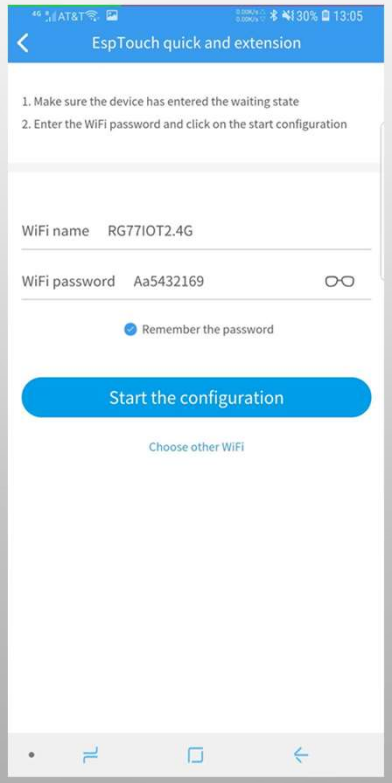
Integrated System (Circuit)

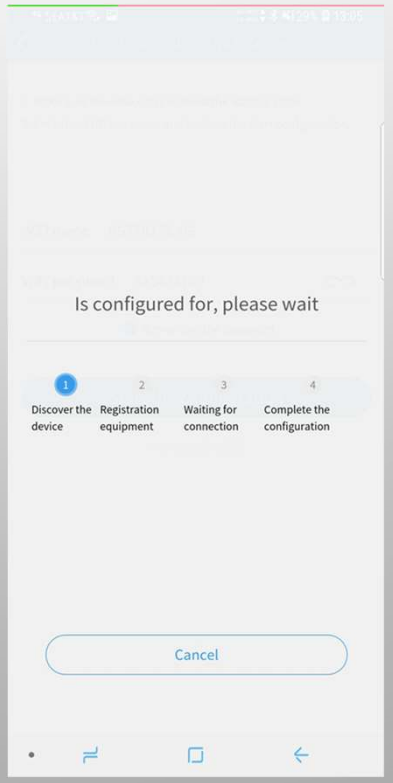


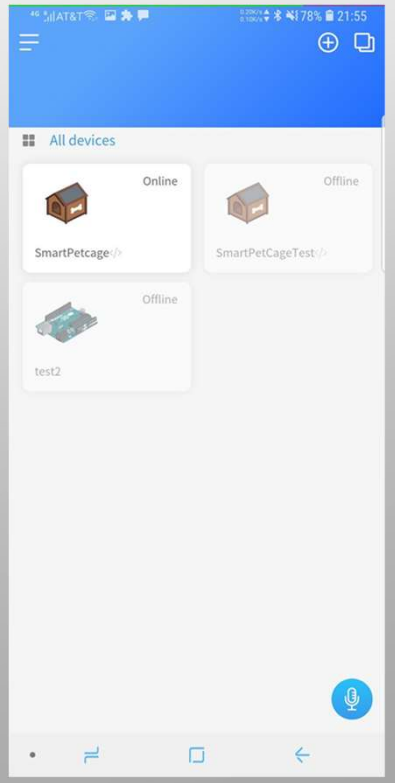


Integrated System (Software)

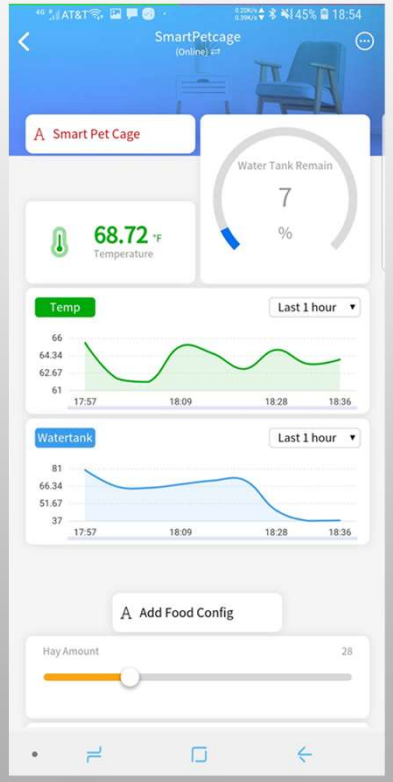


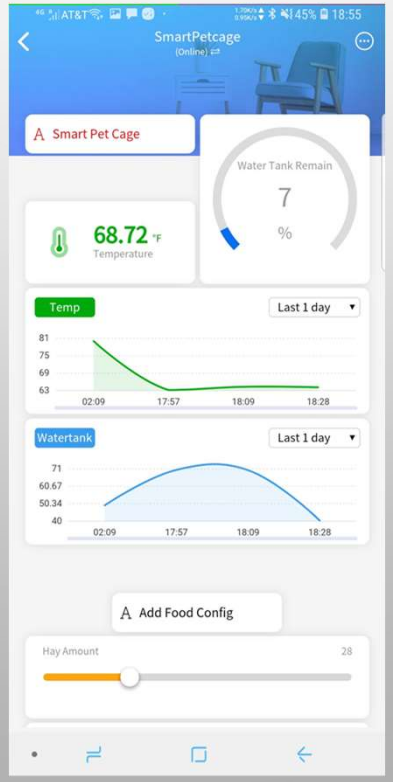


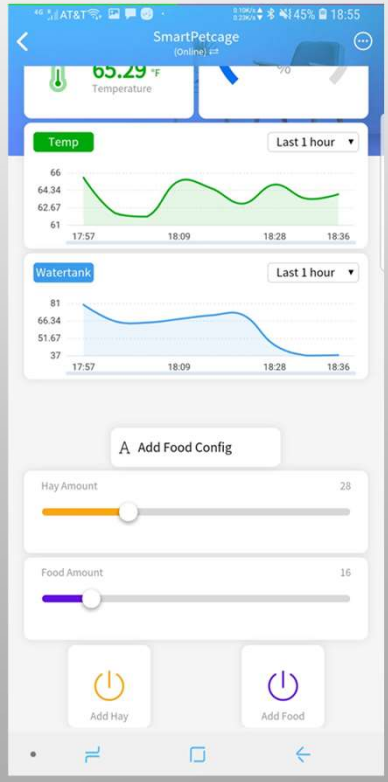




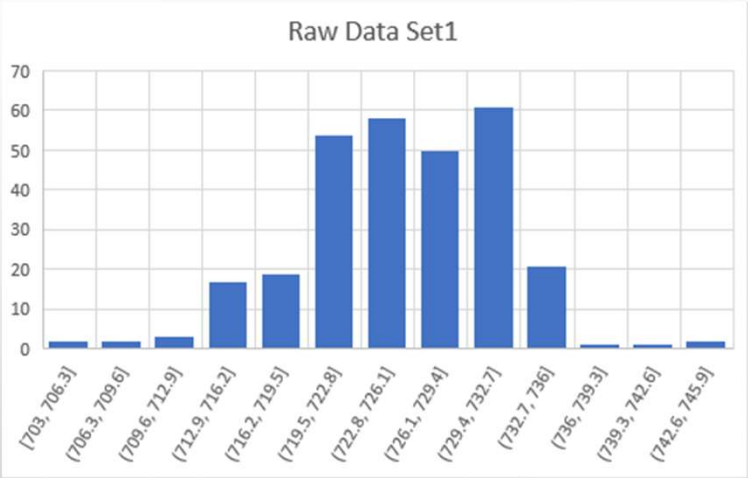
Integrated System (Software)

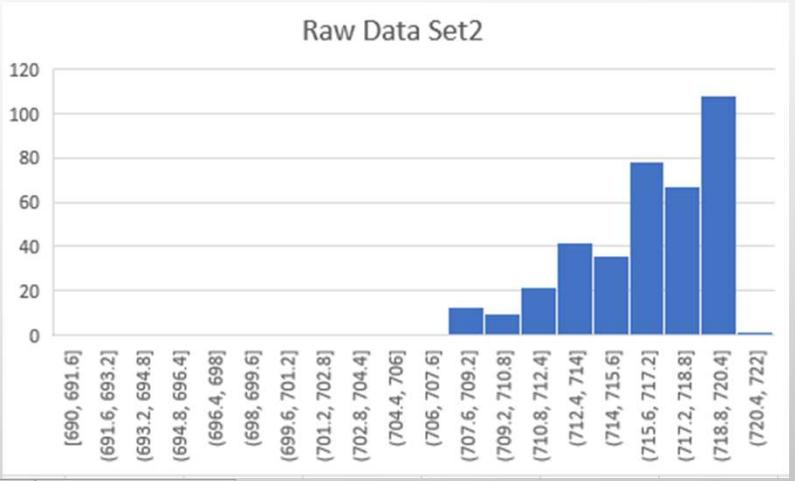






Project statistics (Temperature Sensor Ideal)





# of Data	291
mean	725.3299
std	6.179591
Min	703
Max	743

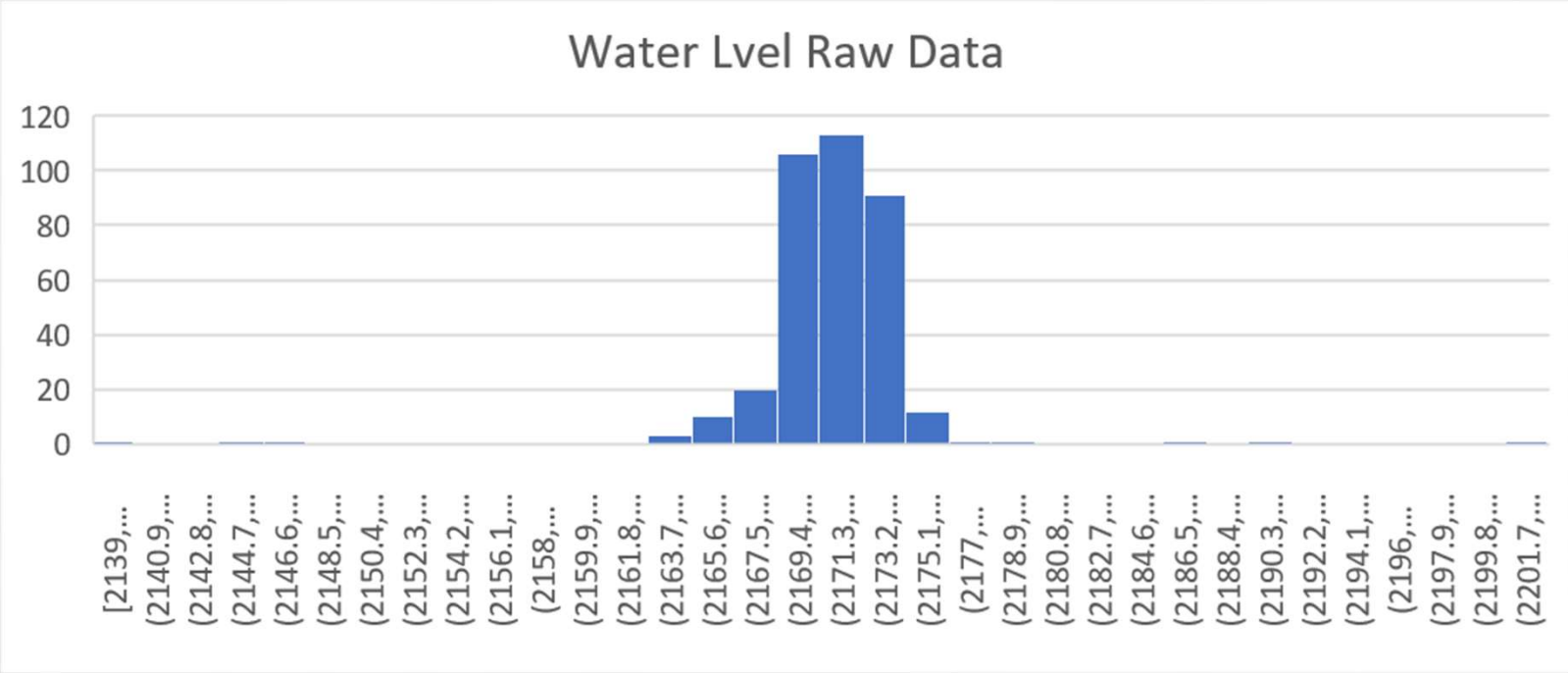
significant level	
0.05	
Confidence Interval	
0.710005	D1
0.340543	D2

# of Data	381
mean	716.346457
std	3.39146207
Min	690
Max	721

Project statistics (Temperature Sensor with 20 data)

Raw Data Set1	# of Data	20		significant level		Raw Data Set2	# of Data	20
725	mean	727.9		0.05		719	mean	717.9
720	std	7.210373		Confidence Interval		719	std	2.36431808
743	Min	715		3.160027	D1	720	Min	712
735	Max	743		1.036189	D2	719	Max	720

Project statistics (Waterlevel Sensor Ideal)

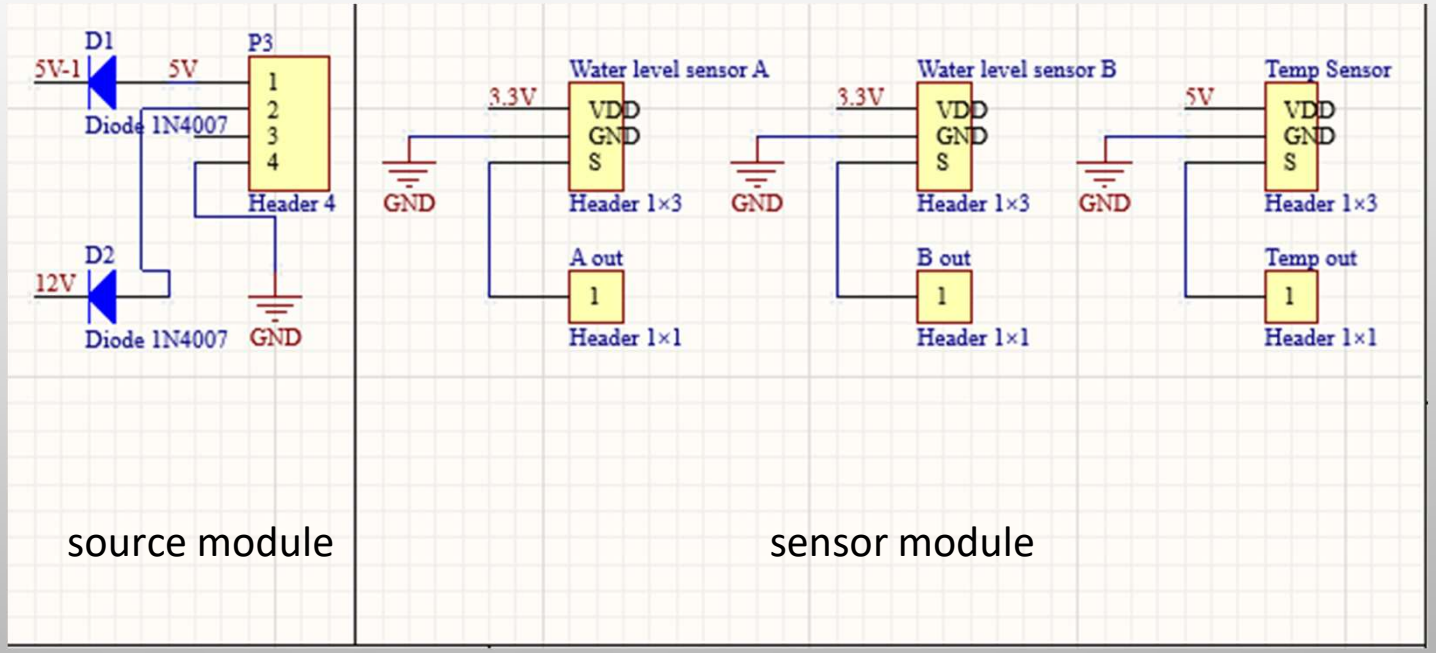


# of Data	363		significant level
mean	2172.198		0.05
std	3.954758		Confidence Interval
Min	2139		0.406831572
Max	2202		

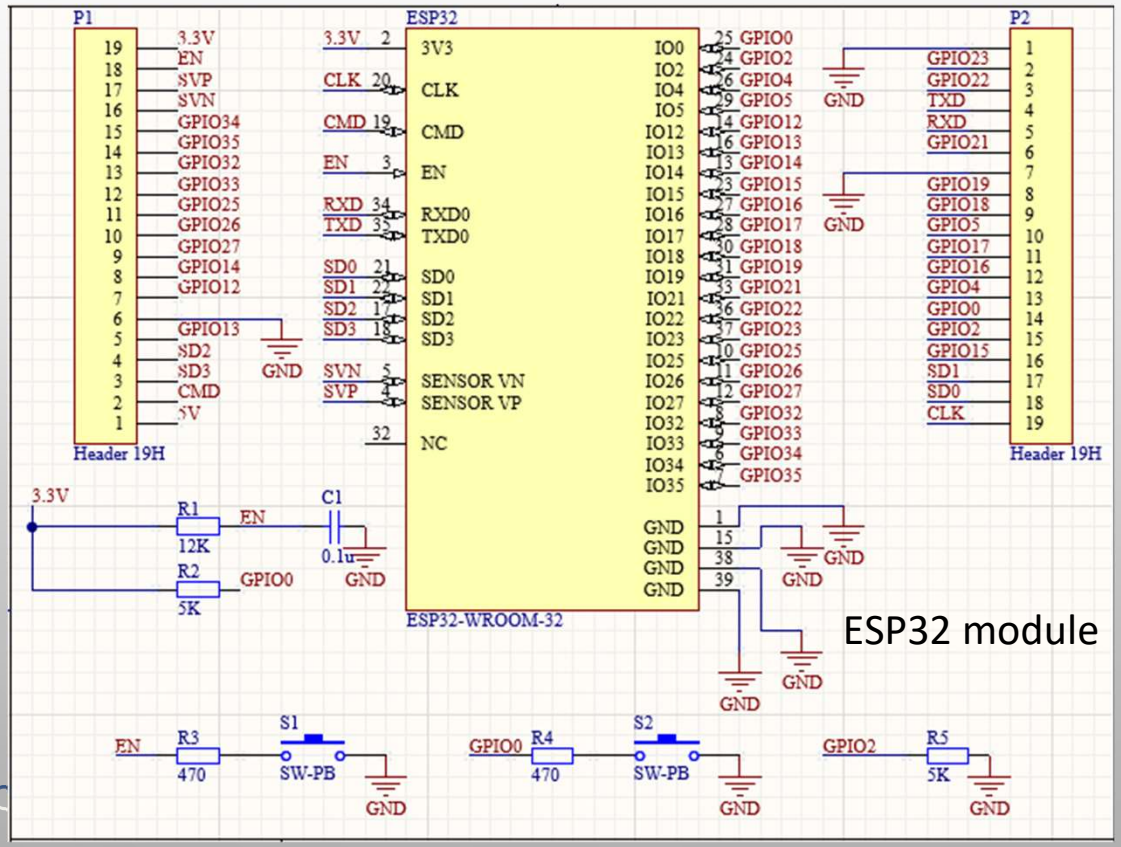
Project statistics (Waterlevel Sensor with 20 data)

RawData	# of Data	20		significant level
2174	mean	2173.85		0.05
2174	std	1.348488		Confidence Interval
2173	Min	2171		0.590990253
2173	Max	2176		

Custom PCB HARDWARE

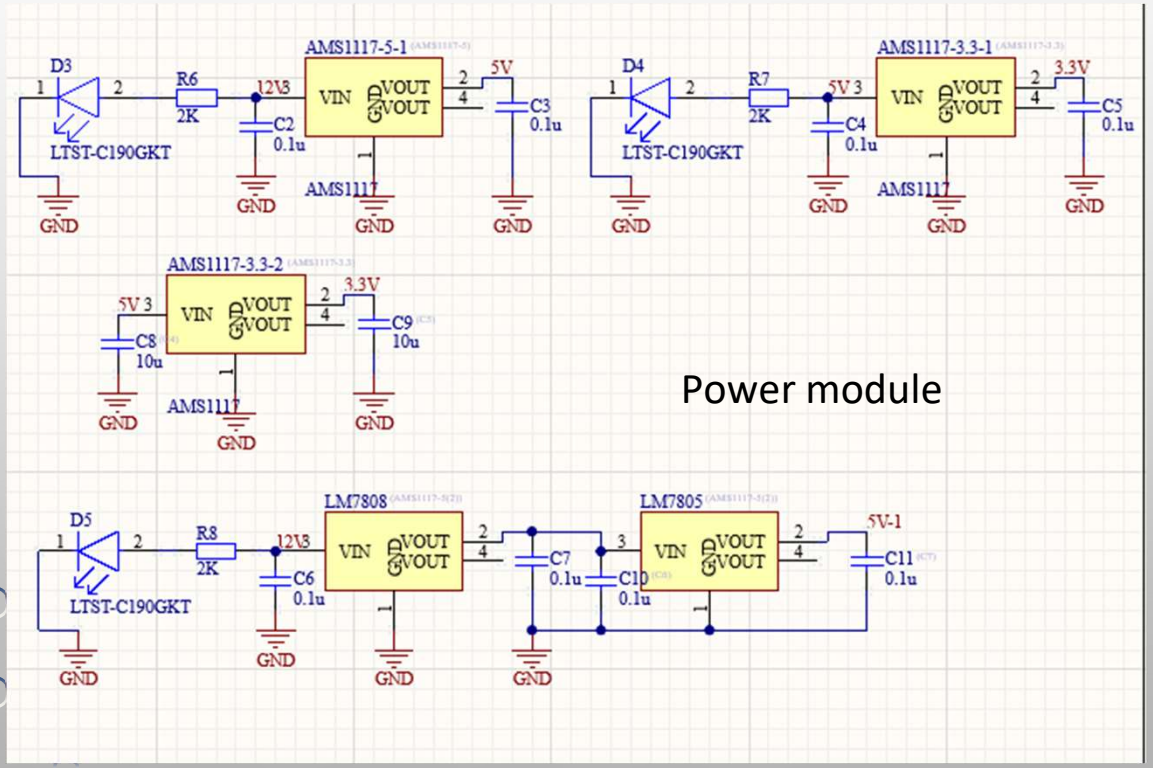


Custom PCB HARDWARE

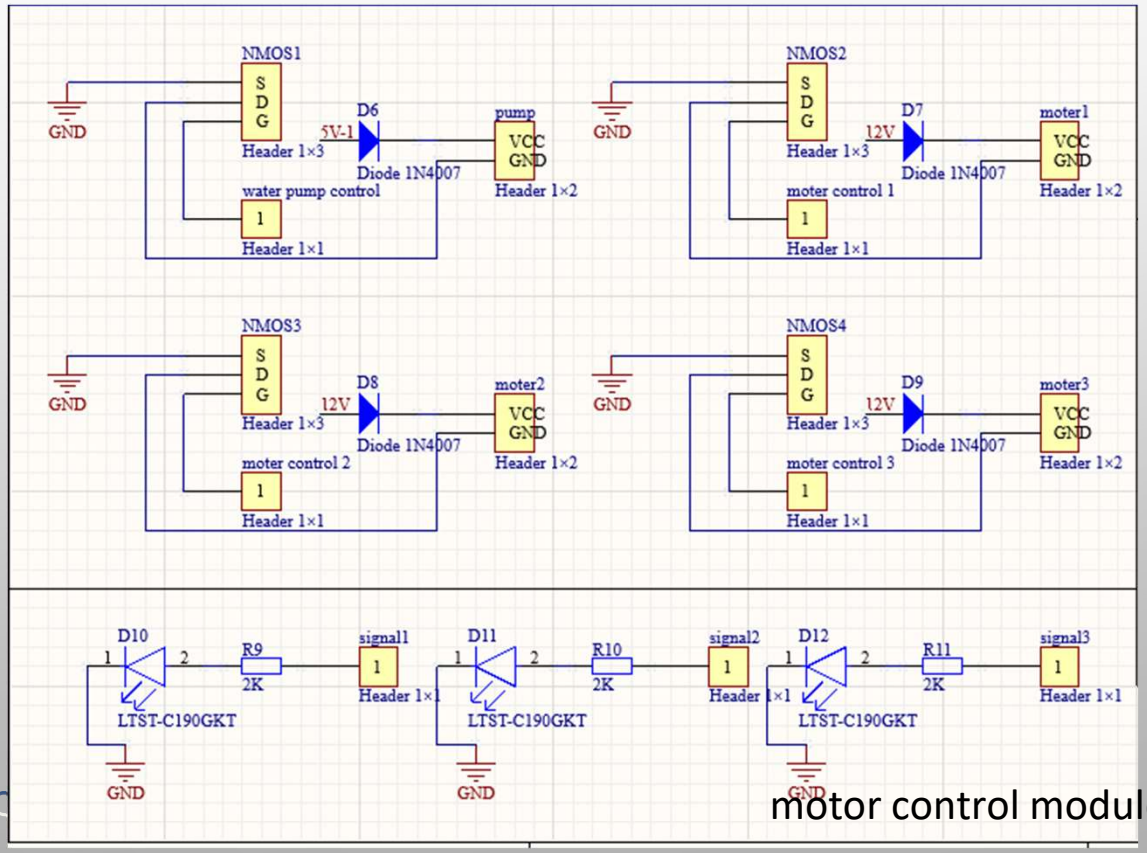


ESP32 module

Custom PCB HARDWARE



Custom PCB HARDWARE



motor control module

UMass Amherst
College of Engineering

Thank You!