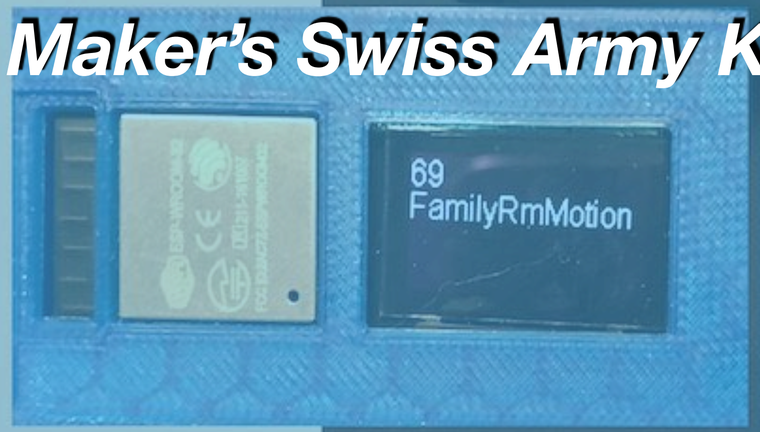
RocketManRC

Maker Faire

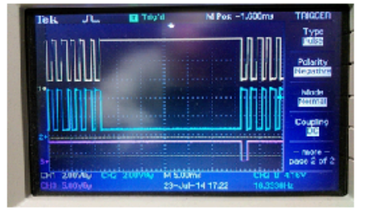
MQTT

The Maker's Swiss Army Knife

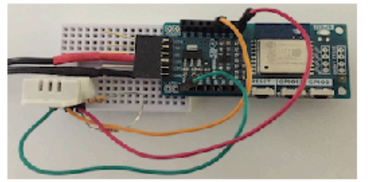


Maker Faire

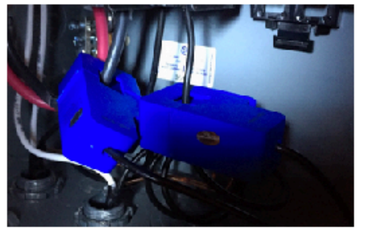
Presentation Objective



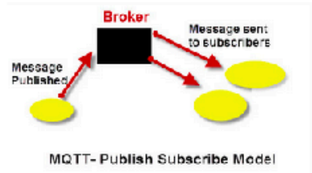
Once you've decided on an approach for your project then you are often faced with a dilemma if you want to integrate it with something else or access it over the Internet!



In this presentation I am going to show I addressed the connectivity issue by using MQTT in my home automation system that has evolved over the past 20+ years.



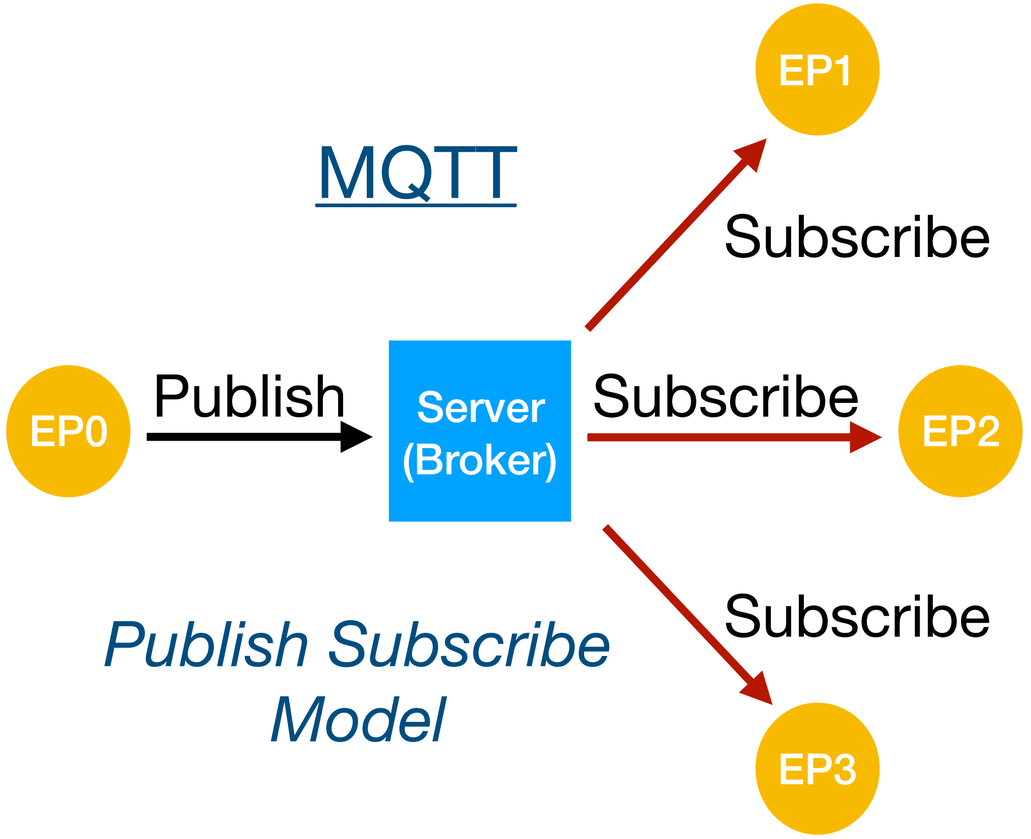
Since this is such a short presentation I have written about six of my projects on my website as a reference:



http://www.rocketmanrc.com

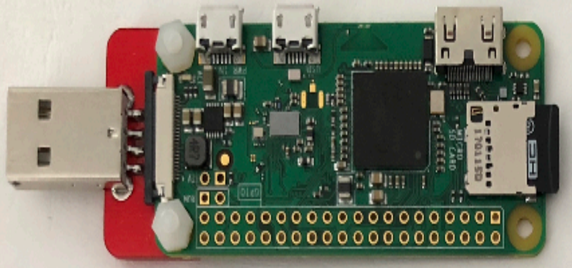
MQTT

- A publish and subscribe architecture over TCP/IP which is very efficient and robust
- A Raspberry Pi makes a perfect server (broker)
- Any micro controller with WIFI can be the endpoint (e.g. ESP32)
- Interface code is minimal, almost any programming language and common dev environments are supported

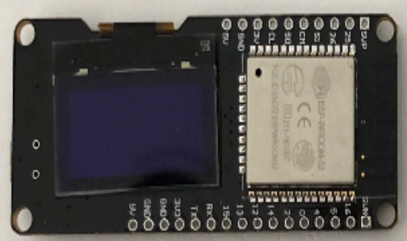


Project Choices for MQTT

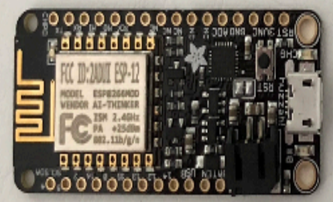
- Processors: Raspberry Pi, ESP8266 & ESP32 boards
- Development Environment: Linux, Arduino, Platform.IO
- Programming Language: C++, Python, Lua, Javascript, etc.
- Software Libraries: Github is your friend!



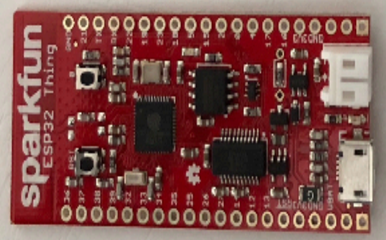
Pi Zero W



ESP32 with OLED



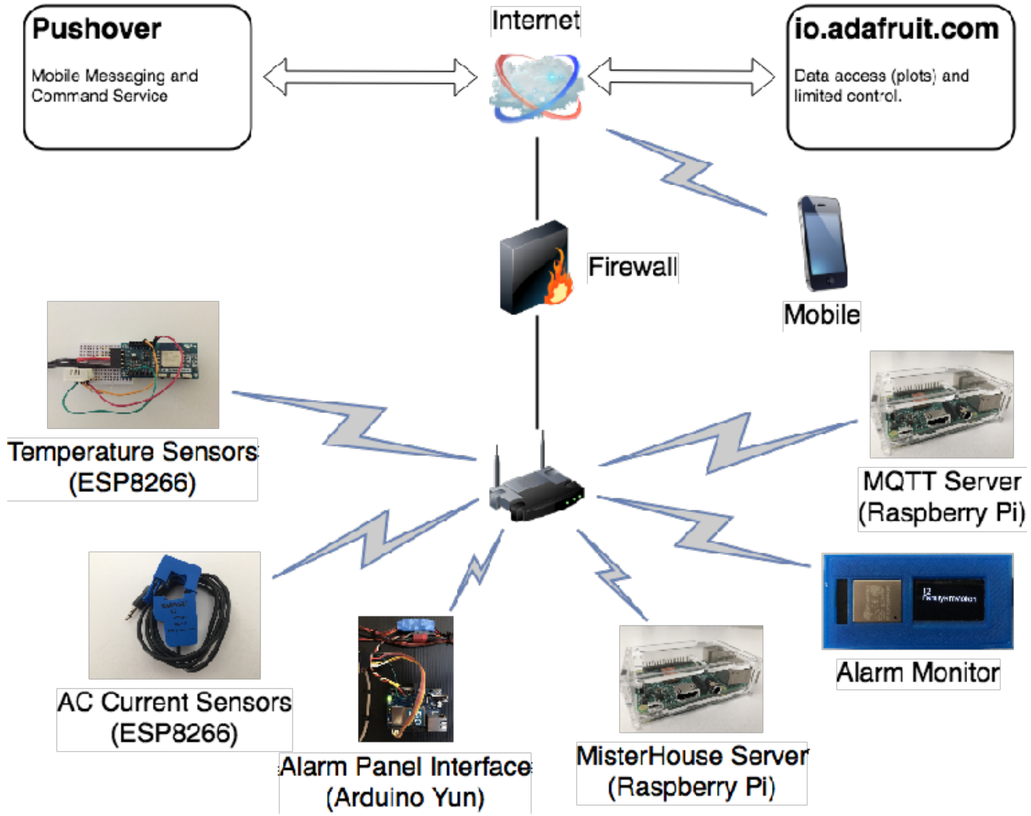
ESP8266 Huzzah Feather



ESP32 Thing

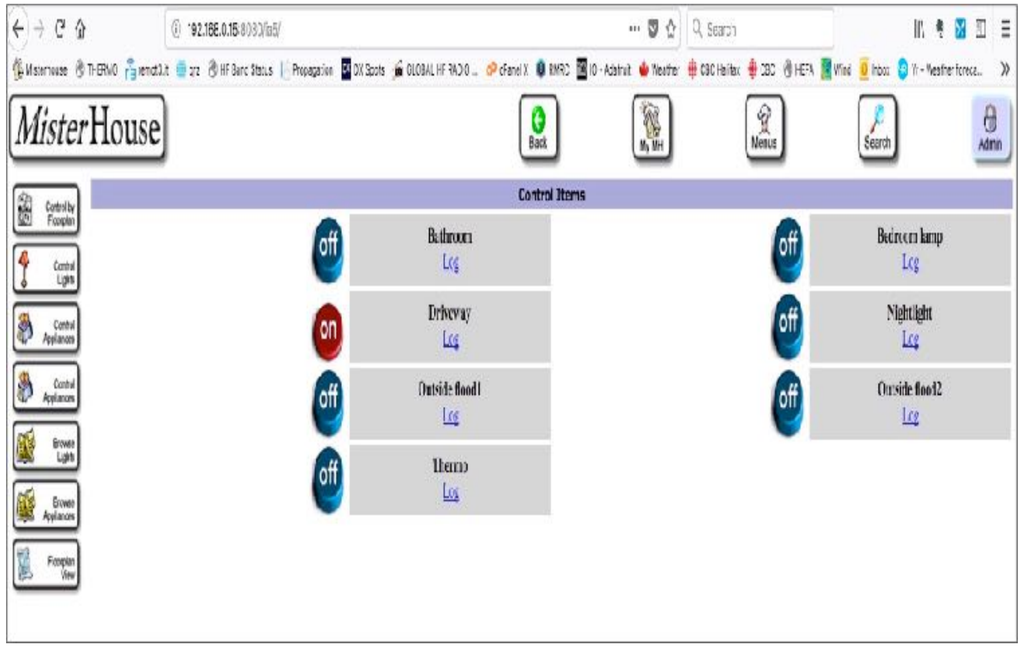
My Home Automation System

- MisterHouse Home Automation Software
- Vintage Alarm System (hacked!)
- Monitoring System for home temperature and heating system electrical performance
- Access to data and some control over the Internet
- Mobile Alarms and Commands



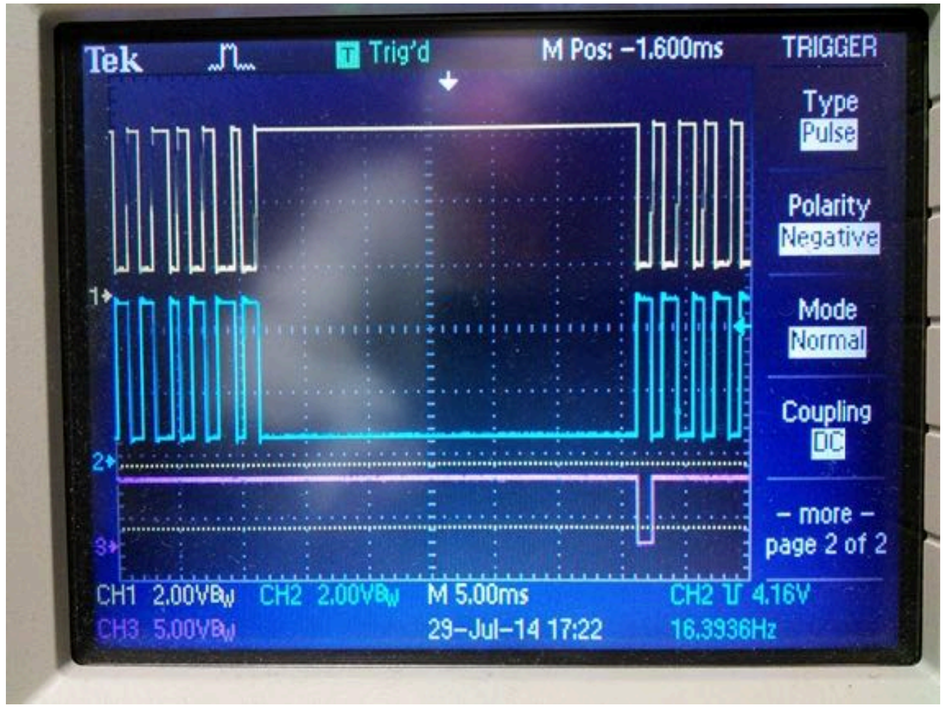
MisterHouse Software

- Started in 1998, written in Perl
- Still in development after 20 years!
- Supports almost any home automation device imaginable.
- For lighting control I use X10, INSTEON & UPB
- Web browser and TCP/IP HTTP interface
- Hosted on a Raspberry Pi (V2)



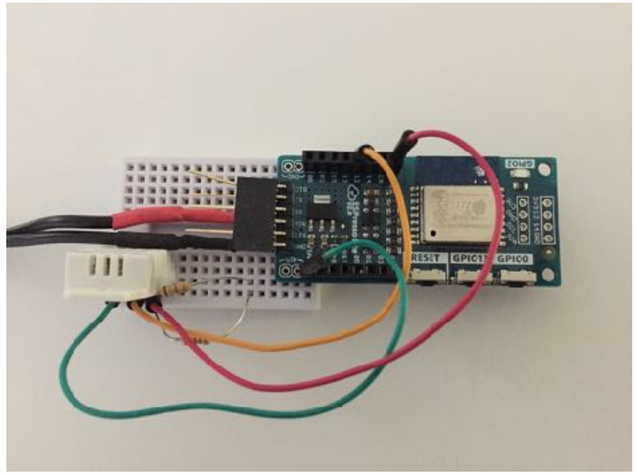
Alarm System (hacked!)

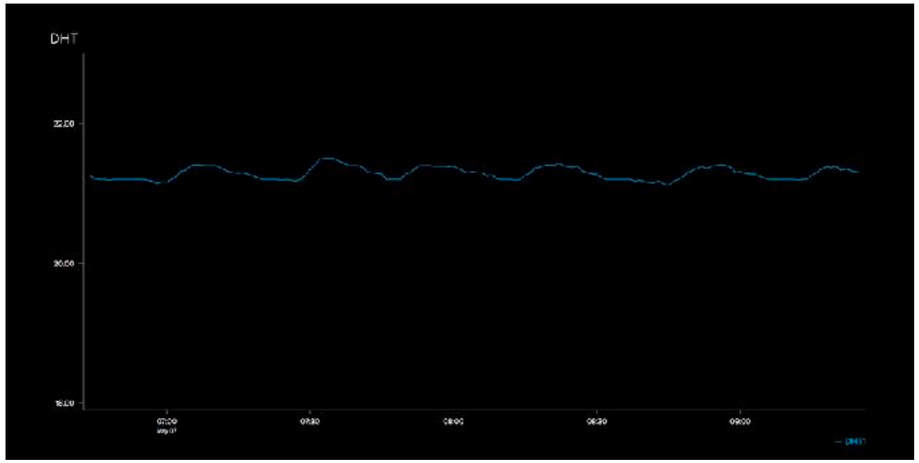
- Had to unlock the panel to change the phone number
- Reverse engineered the keypad databus
- Used an Arduino to enter the 10,000 possible lock codes
- Later used an Arduino Yun to interface to a Raspberry Pi server running Mosquitto
- First use of MQTT!



House Temperature Monitoring

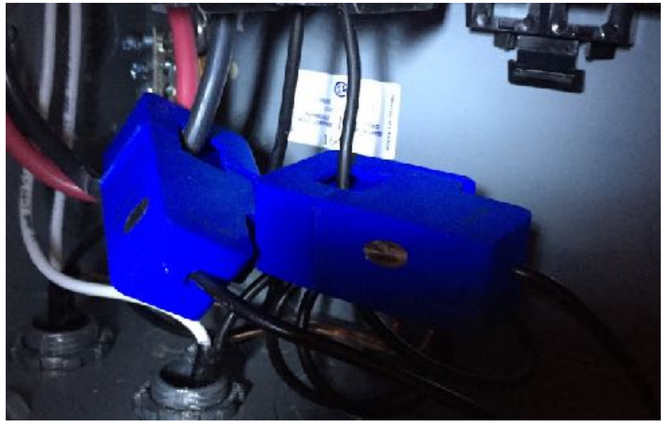
- Essential for early detection of heating problems
- Temperature (and humidity) measured with DHT22 sensors
- One sensor on a heat register and the other in the kitchen
- Data is transferred directly to io.adafruit.com via MQTT for remote access (displayed as a graph over time)
- Board is a ESPresso Lite V2.0 (ESP8266)





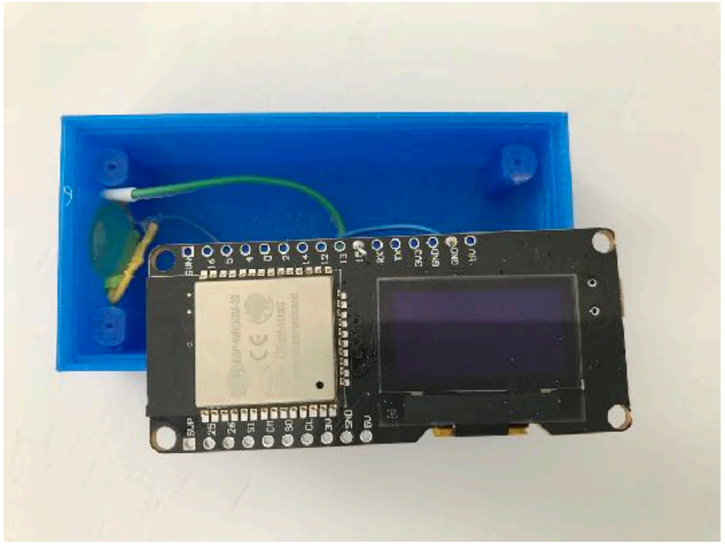
AC Current Monitoring

- Used to monitor Heat Pump and auxiliary heating system performance
- Inexpensive clip-on sensor from Seeedstudio
- Uses a library from the OpenEnergyMonitor project (EmonLib) for processing
- Data is transferred directly to io.adafruit.com via MQTT for remote access (displayed as a graph over time)
- Board is a ESP8266 Huzzah



Alarm System Monitor

- Bedside monitor to display alarm conditions (alarm system can be unarmed)
- Integrates to alarm panel (MQTT) and lighting system (HTTP URLs)
- UI is a tilt sensor
- Board is a Wemos Lolin32 (ESP32) with OLED
- This project took only a few hours from concept to implementation!



The Alarm Monitor Code (defs)

```
#include <PubSubClient.h>

#define FEED_PATH “dsc/panel"  // MQTT topic

void callback( char* topic, byte* payload, unsigned int length );

WiFiClient wifiClient;
PubSubClient client( server, 1883, callback, wifiClient );

void callback( char* topic, byte* payload, unsigned int length )
{
  Serial.println( topic );
  Serial.write( payload, length );
  Serial.println( "" );
}
```

The Alarm Monitor Code (setup and loop)

Add to setup():

```
if( client.connect( "ESP32Client", "", "" ) )
{
    Serial.println(F( "MQTT Connected" ));
    displayStatusMessage( (char *)"MQTT Connected" );

    int err = client.subscribe( FEED_PATH );
}
```

Add to loop():

```
client.loop();
```

Operation

Questions?