

Home Alone Monitor

Project Description

Peter K. Boxler

January 2020

Table of Content

1. Abstract	3
2. Introduction.....	3
3. What does it do.....	4
4. Parts used	4
5. Software	4
6. Finite State Machine.....	5
7. Multitasking ESP32	5
8. Oled Display	6
9. Reporting of Movements	7
9.1 Thingspeak Channel.....	7
9.2 Pushover Messages	8
10. Code	8
11. Discussion, Conclusion	8

1. Abstract

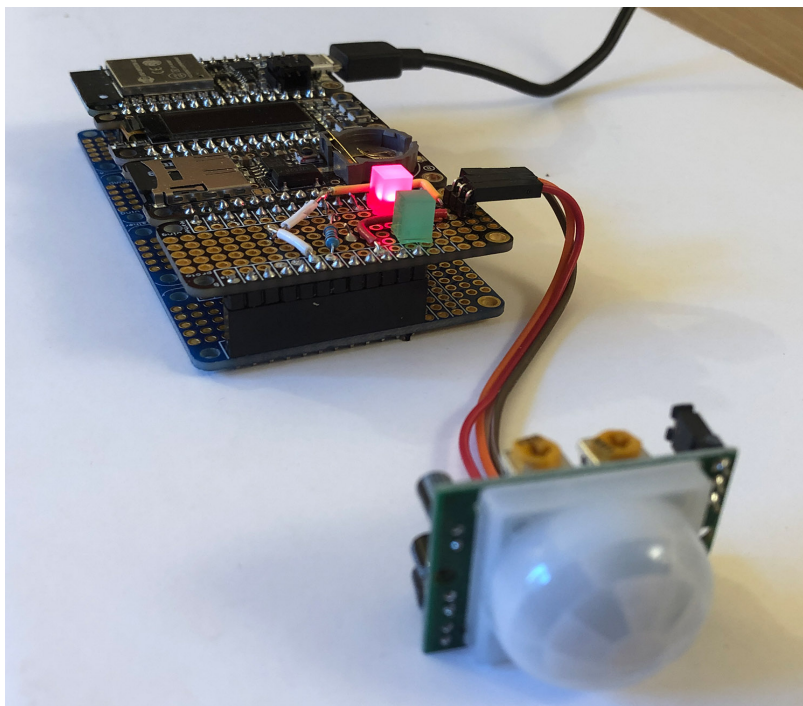
This documentation presents a small electronic gadget called Home Alone. Some folks concerned with the well being of elderly people living alone could use it to keep track of the movements of their loved ones. This gadget detects, counts and reports movements in its surroundings. For privacy reasons no camera is used, a simple PIR sensor (Pyroelectric Infrared Sensor) detects movement. Movements (or the absence of it) are reported to the cloud. Concerned party (or parties) can inspect the data on a website.

2. Introduction

Recently a friend told a story of an elderly woman being found dead by her daughter. Luckily, the daughter came by just one day after the old lady passed away. Everybody has heard of similar stories. That got me thinking and I knew it would be rather easy to build a gadget helping to prevent such stories. However, a couple of days later I discovered that (here in Switzerland at least) one can buy electronic devices that detect movement and they are connected to the landline. So I dropped the plan to build such a device.

A couple of weeks later, however, I came across [Ralph Bacons Maker Videos](#) on YouTube and bingo, he describes just such a device in one of his videos.

Since most of the parts needed to build such a gadget were lying around in my lab I decided to give it a try. Here is what came out of it. It is just a prototype as of now.



Home Alone Monitor Prototype

3. What does it do

Let's assume we have an elderly uncle called uncle Dagobert. He lives alone in an appartement in a city a couple of hours away. He still can live on his own, is even able to walk to the shop or to the city park and he can handle a phone. His neighbours keep an eye on him and they promised to call us if they have not seen him in days.

So this is where this device comes in handy: it detects movements (in the appartement) and counts them. Movement counts are reported to the cloud in defined intervalls (every hour or so) and can be inspected on a smartphone anywhere in the world. The provider Thingspeak is used for that. At least once a day (in the evening) a push-message is sent to a smartphone reporting the movement count during the day. So even if we forget to check the Thingspeak website we will have a message on our phone. All of these thresholds and intervalls are configurable - the json configfile resides on a micro-sd card. This scheme gives us some reassurance that all is well with uncle Dagobert (or not, see discussion at the end of this document).

4. Parts used

I love the Adadfruit Feather series of boards and therefore decided to build with what I already had. Ralph Bacon device uses an ESP8266 but I was keen on using the more modern ESP32.

So I assembled these Feather boards for my device:

- Feather Huzzah ESP32, Product ID: 3405
- Featherwing OLED 128x32, Product ID: 4091
- Adalogger Featherwing, Product Id: 2922
- Featherwing Proto, Product ID: 2884
- Adafruit Quad Side-by-side, Product ID: 4245
- PIR Sensor, Product ID: 189
- Two LED, resistors
- Micro-SD card (any size will do)
- Power Supply 5 Volt 1 A

5. Software

The nature of the problem calls for an implementation of a Finite State Machine. Uncle Dagobert, whose movements are to be detected/reported can either be at Home, Leaving the home or be away. I choose to adopt Ralph Bacons idea that uncle Dagobert should press a button before leaving the appartement.

Frankly, however, I suspect that in most cases uncle Dagobert simply forgets to do this. So there is a great deal of interpretation when one looks at the movement data reported to the cloud.

Note: at the outset I also had a state NIGHT during which no reporting was done. After a while, I felt that this is not really needed.

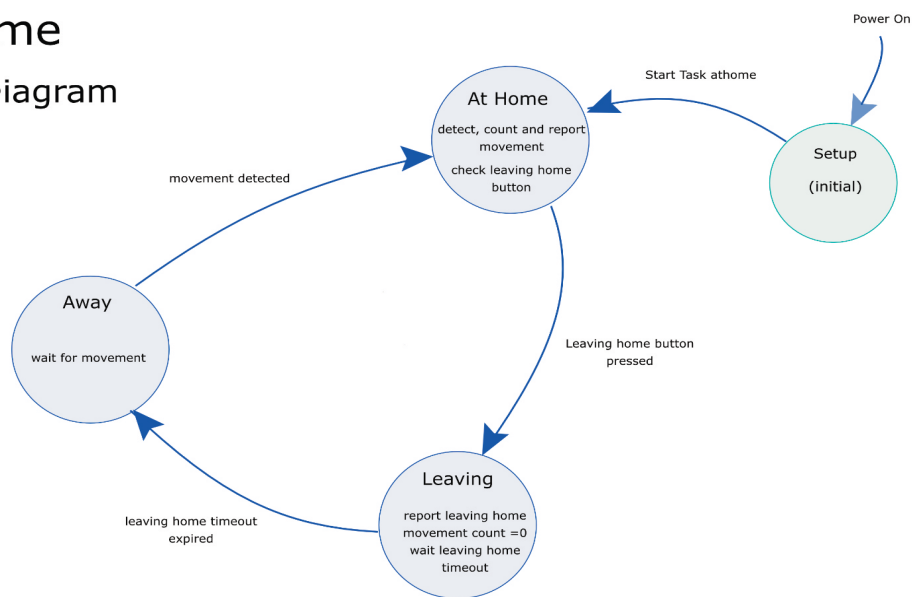
I also wantd to make use of the multitasking capabilities of the ESP32 chip.

I used the Arduino IDE to develop the code and for readabiltiy and beauty I arranged the code into a couple of modules. Ralphs code is hard to read - all in one long piece.

6. Finite State Machine

This the state diagram

Alone At Home State Machine Diagram



Version 1
Peter K. Boxler, Jan 2020

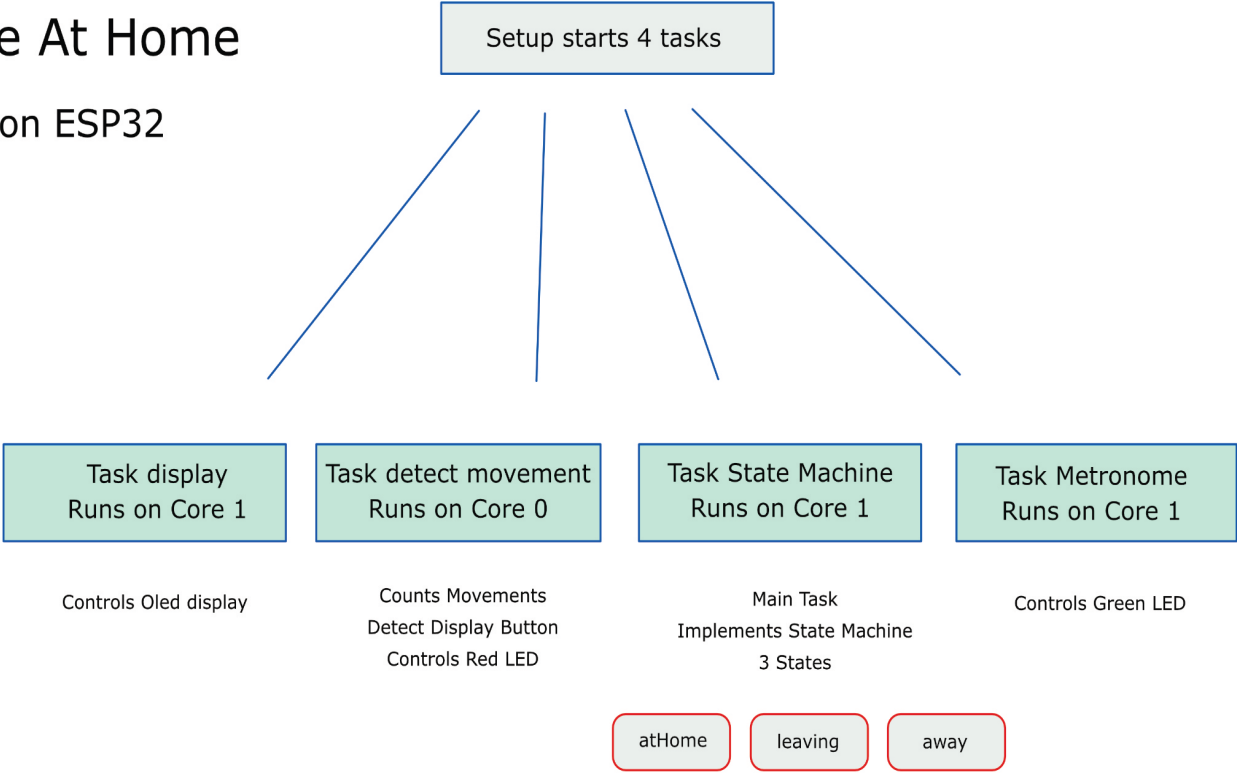
State Diagramm of the application

7. Multitasking ESP32

This chip has two cores, runs with 240 Mhz clock and there is even a Real Time Operating System (RTS) on board. My code makes use of these capabilities and the setup function (arduino speak) starts 4 concurrently running tasks.

Alone At Home

Tasks on ESP32



Version 1
Peter K. Boxler, Jan 2020

ESP32 Tasks

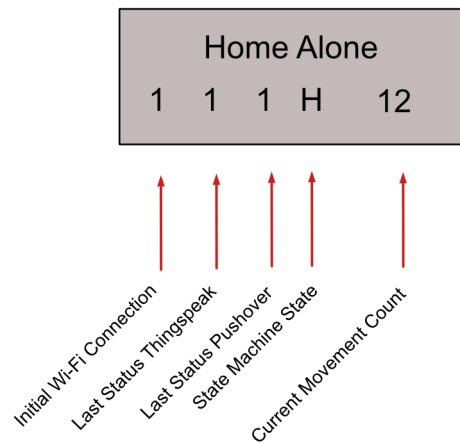
8. Oled Display

A small oled display shows just a few simple internal states of this gadget - the display is kept simple and a large font is used for easy reading. These values are shown (1 meaning all is well):

- Status of the initial wifi connection (during setup),
- Status of the last reporting to Thingspeak.com
- Status of the last message to Pushover.net
- Current state state of the state machine
- Current movement count

This is the old display:

OLED Display

*OLED display Home Alone*

9. Reporting of Movements

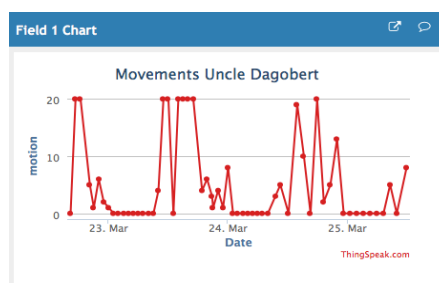
9.1 Thingspeak Channel

The movement reporting is done with Thingspeak. The reporting intervall is configurable, I used 80 minutes.

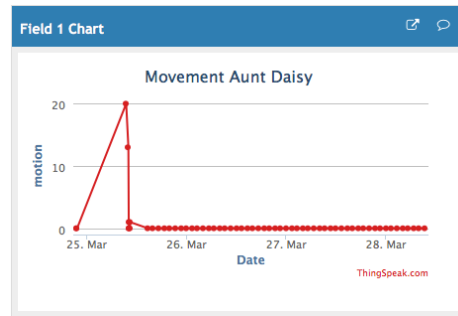
The test-channel is public and can be found here.:

[Uncle Dagobert Home Alone](#)

Example: the following picture shows a regular report of Uncle Dagobert who seems to be alive and well:



Aunt Daiy, on the other hand, seems either absent from home or dead:



Reporting with no movements during 3 days

9.2 Pushover Messages

Instead of the sending of emails I opted to used **Pushover.net** to send messages to mobile devices. There is a **morning reporting**, an **evening reporting** and an **alert reporting** if no movements are detected within a configurable length of time.

Morning/evening reporting are configurable: one can have both, one or none of them. They report the number of movements since last time (the reporting was done.). Morning hour and/or evening hour are configurable (zero meaning no reporting).

If there are no movements during a configurable length of time (I use 24 hours) are detected, an alert message is sent: '**Attention: No movements within the last 24 hours**'.

10. Code

The code is available on GitHub.

11. Discussion, Conclusion

When uncle Dagobert shuffles around in his appartement and his WLAN is up we habe no problem: we can see that he is alive.

But what do we really know if no movements are reported to the cloud - this is the tricky part.

- The WLAN might be dead in his appartement for a multitude of reasons unconnected with Dagobert himself.
- Thingspeak as a company has folded or their computers have a serious problem.
- He might have died in his sleep last night
- While shopping, he might have met this lovely old lady and they ran off together
- Dagobert might have tripped over the power cord and the device is dead
- The ESP32 might have died but Dagobert is still alive and well

We conclude that this device is nice and can help us to some degreee but....

All in all, we are happy we have it in place in uncle Dagobert's appartement.

end of document