Automated Home  
security system

# Introduction

The project I have chosen to do is a smart home security system. When I was smaller I loved watching Home Alone every Christmas, and the scene where the intruders knew that no-one was home, by timing the automated lights gave me inspiration for this project.

Home automation is the use and control of home appliances remotely or automatically. The concept was first made and experimented with to save labour and time. Some of the devices used can communicate with each other and so are part of a term called IoT. This term will be discussed and explained later. Others are stand alone systems can can only be programmed at the machine, not via a remote of any sort (like a TV remote or another computer). Some home automation systems that you may not thing are categorized under this definition are dish washers or even an oven or microwave.

Machines that can communicate with each other, or can be programmed from afar are part of the “Internet of things” or IoT for short. There are many forms of communication used, from simple R232 serial communication, to infrared, to Bluetooth and finally the most obvious one Wi-Fi. Some of the items only have pier-to-pier connection (2 machines links directly and no others can be added), some have a wide web like communication and there are levels in between. It is a vast advancing field area were only the imagination creates boundaries.

However nothing is flawless and the more systems and apparatus linked and communicating with each other increases the chance that it will be the target of a cyber attack. This becomes an even larger problem when the systems are not just linked with each other, but can be accessed from the outside world.

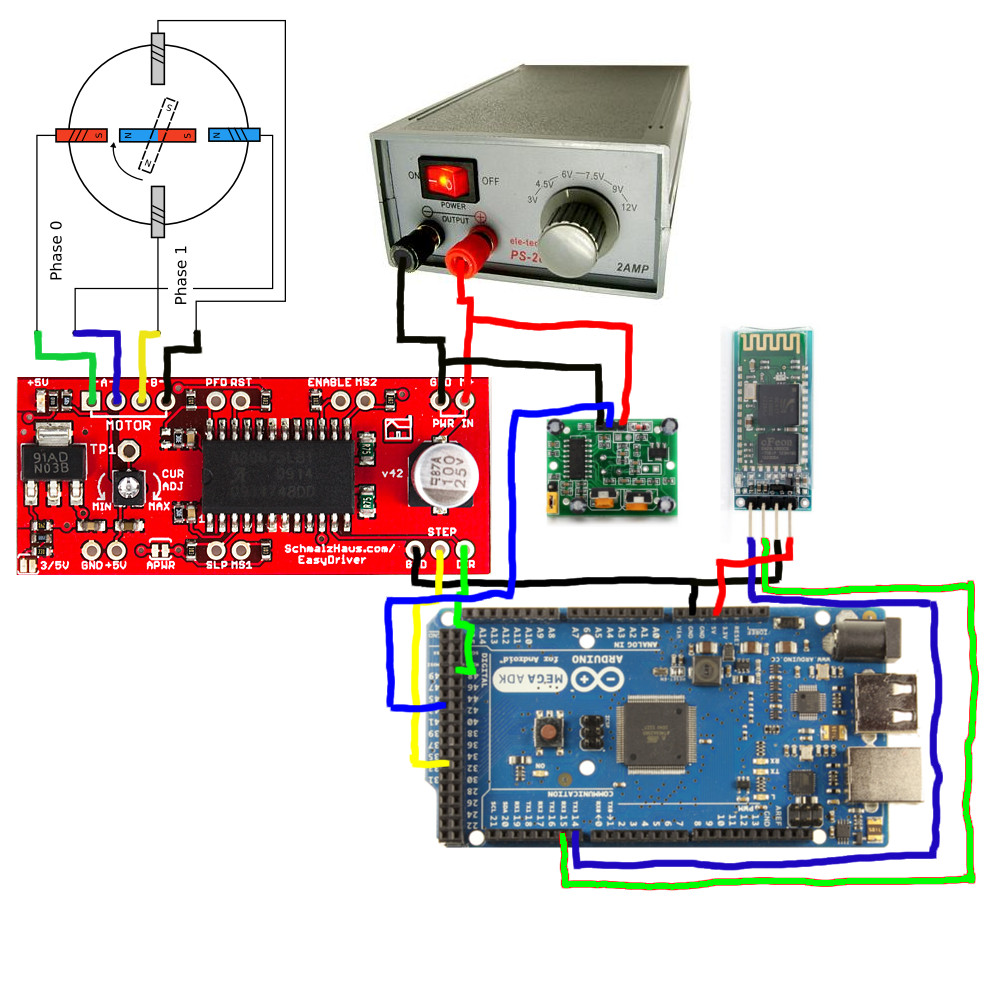
Furthermore IoT is still in the infant stages of its life and has limited equipment available for users and developers to choose from. This was one of the biggest issues for me while doing my project as well. In addition everyone is inventing different protocols which usually do not communicate with each other. Lastly to add to the negatives the cost of installing and expecially maintaining the instrument has a large cost, however this is the fact with innovations and hopefully with the expansion of the market and the supply and demand stabilizing this will change.

After all the negatives I want to point out that there are many positive aspects of this technology. Apart from the fact that it is extremely conformable being able to access any device from any place, there are security and economically benefits to this technology. First of all, think about a scenario when the family is aboard on holiday, and they are robbed. They will not know about it until they are home, and even if they had an alarm installed and the neighbors were home and called the police, they would not be able to determine what was missing. If they could access the alarm from any place on earth. An sms or email could be sent to their phone and the authorities in that precise moment when the intruders entered the house. As well as that extra locks could be activated and the family members could identity the missing items and the authorities had more time to retrieve them.

Furthermore the economical effect are greatly beneficial. In the case a person leaves his home or the room they were previously occupying and forget to turn the lights and other electronics off, a built in motion sensor and timer could automatically do this for him/her. With added performance/power monitoring equipment, the exact amount of electricity used could be calculated and cut/lessened if needed. The sprinklers in the garden could be set on a timer and not left on overnight and so forth.

**IoT and home automation used in My Project**

The main concept of my project is much like many other home security systems. When the alarms are triggered, a signal will sent to the central unit and an alarm will sound. The difference between this and other systems is, that the motion detection sensors are placed on all openings of the house or building meaning the alarms will be triggered at the moment of entry not allowing time for the signal to be shorted by the infiltrators. Moreover I will not be using wires meaning there is nothing to physically cut.

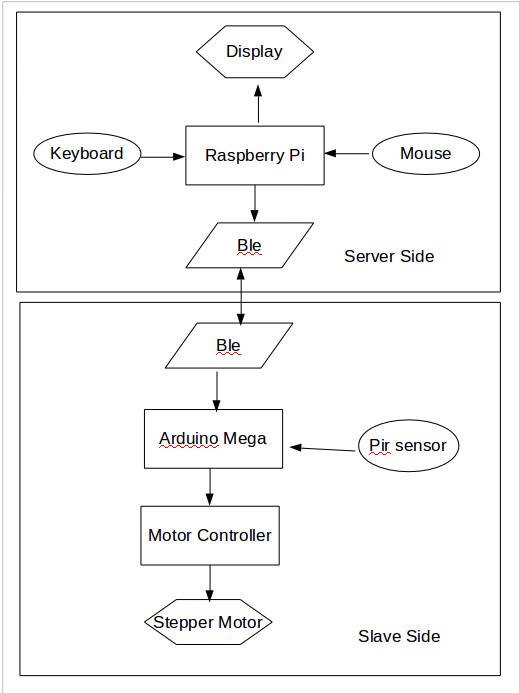
**Main Features**

* Wireless communication (bluetooth)
* Master-Slave setup and communication
* Automatic Motor movement

There are three main features of my project, some of which have been mentioned beforehand in this document. Feature number one is that all sensors and motors will be wireless and will communicate with a central unit (called CU from now on). All sensors and motors will be linked to each other on an Arduino Mega 2560 microcontroller which will in turn be linked to a Raspberry PI (CU) via Bluetooth communication.

The next feature is that the user uses a master unit called Central Unit (or CU for short) the give commands and operate the elements linked on the Arduino board, like the PIR sensor and the motors. The user will see messages and status reports on the CU via a serial monitor.

The last feature that will be implemented in this project is that the user can set a morningAlarm() and eveningAlarm() time and each day at these times the blinds will be opened or closed.

**Inputs/Outputs(I/O)**

Block Diagram featuring I/O's

Slave side physical I/O implementation

The inputs of my system would be a manually set time, set by the user. Further inputs would be the signals from the motion sensors. Outputs would be the motor closing the blinds and a buzzer sounding if the motion sensors were triggered.

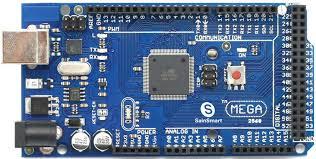
**Used Hardware/Software**

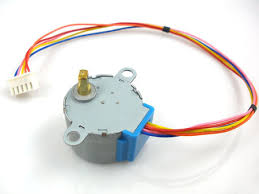
* Raspberry Pi B module
* 5V 4 Phase stepper motor
* ULN2003 Driver board
* Easy Driver stepper motor board
* HC-05 Bluetooth module
* Arduino Mega microcontroller
* König CSBLUEKEY200
* Computer tweeked power supply

The Raspberry Pi is the core of the master side system. All info will run to it and all commands will be sent from it. It will be the only device interpreting the data, and deciding what to do with the given data. For future data storage and so the flash memory is preserved, a external HDD will be installed on the device. The Arduino Mega will be the microcontroller used for interpreting the commands coming from the Raspberry PI via a HC-05 bluetooth module. It has a C based, self written code complied on it, with extra arduino library functions. This element will give the appropriate signals to the motor controller module and transmit the signals from the PIR sensor. The motor will be one of the output devices of the automated system and will be used to drive the shades. When the inputed time of day has arrived, the CU will send a signal via the BLE module to microcontroller and that in turn to the motor controller board, giving the number of steps and the direction the motor must turn (one step equals a 360° turn). The motor will be linked to the motor controller and the motor controller will be connected to two separate GPIO pins of the MEGA board. The microcontroller has many analog and digital I/O pins as well as PWM's and communication pins. We will only need to used 5 digital pins (2 for the motor and 1 for the PIR sensor) and we will also use the GND and the 3.3V and 5V pins of the microcontroller to give the correct voltage to the elements. The PIR sensor needs 3.3V and the motor controller needs 5V. The controller also needs a 12V voltage which an external generator will provide. This is a computer power supply. The PIR sensors are the other input devices of our system. They are used to detect motion in different areas of the house, when the occupants are out of the house. The PIR sensor sends a signal to the TX3 pin of the microcontroller when the device is triggered and this triggers an alarm. Here it is simulated by a smaller buzzer. The computer twitched power supply is a 12V supply from an old Personal computer and the output wires have been modified to be able to connect to a breadboard.

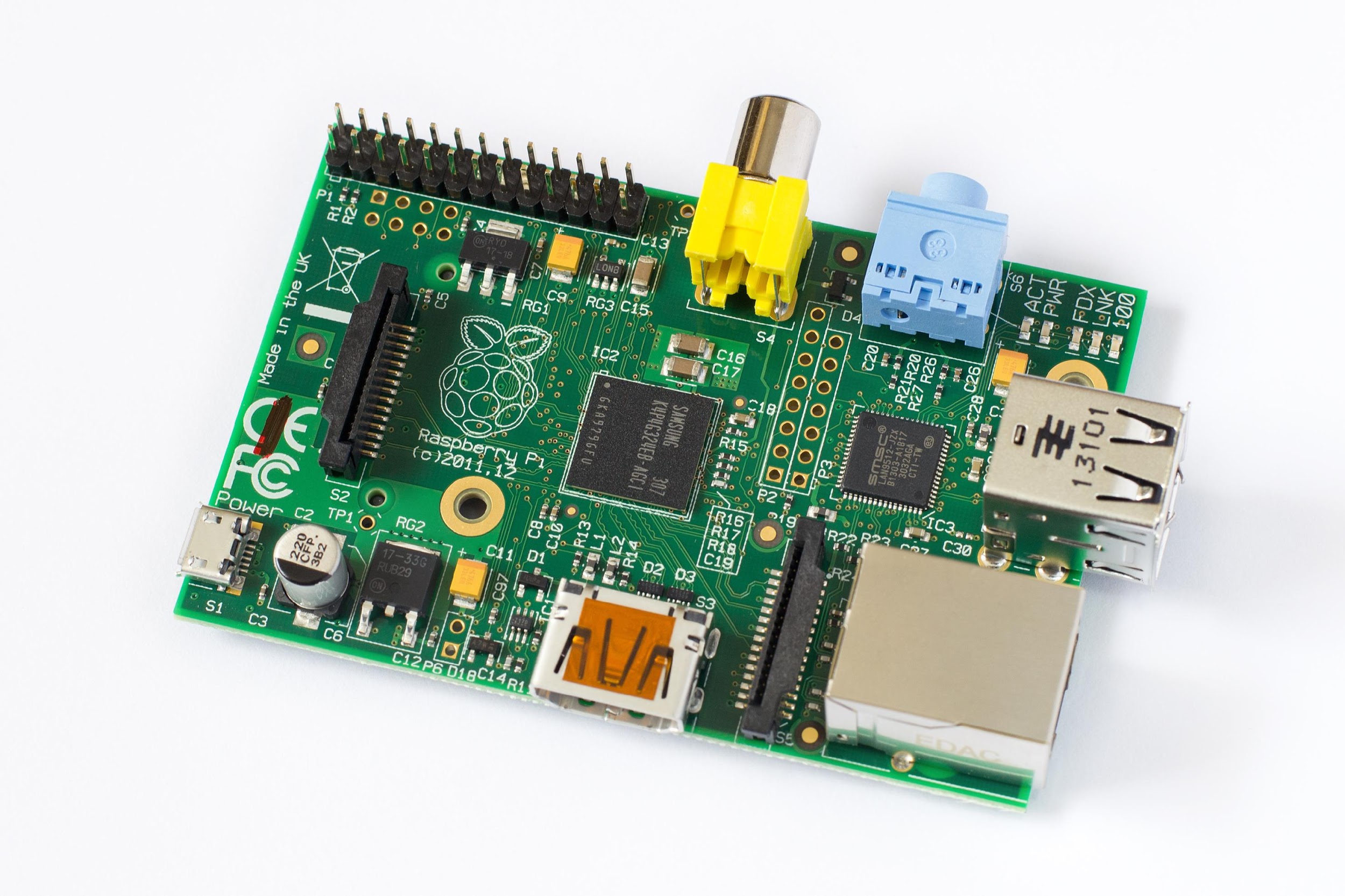
Hardware

1. Arduino Mega 2560



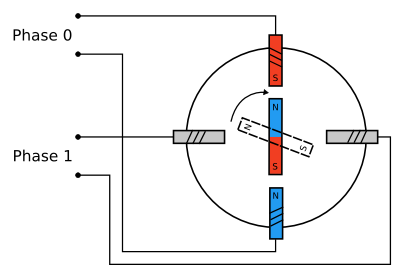
The arduino mega is based on the Atmega2560 which arduino has purchased. The microcontroller has designed for larger projects, as it has 54 digital and 16 analog I/O pins as well as 3 separate communication channels. This board operates on 5V and has a clock speed of 16MHz. The mega can give of a max of 20mA per pin and has a bootloader preinstalled on it. It was ideal for me, as I didn't want to waste time with board configuration.

1. Raspberry PI B module



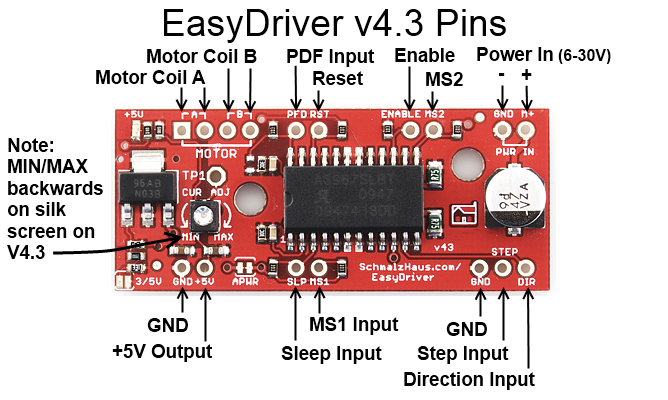
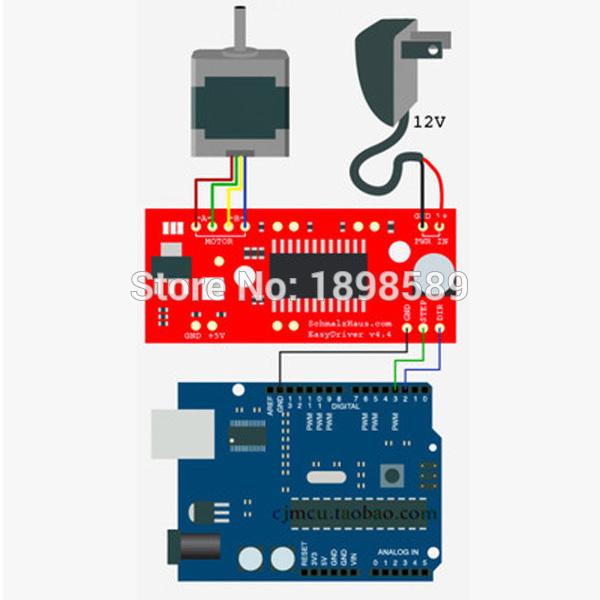
This is a first generation raspberry that has been improved. It has ethernet connectivity as well as a HDMI and two USB 2.0 slots. The amount of USB used was not enough for me, so I added a 4 slot USB hub. This is used for the Keyboard and mouse as well as the Bluetooth dongle. The PI also has 24 GPIO pins, which I did not use, as the PI is used for server side (master role).

1. 5V stepper motor



The reason I chose a stepper motor was for the 360 degree turning capabilities and the easy control. The concept is that there are 4 electromagnets and a normal magnet in the middle. This helps divide the full motion into equal steps. The motor's position can then be commanded to move and hold at one of these steps without any feedback sensor (an open-loop controller), as long as the motor is carefully sized to the application in respect to torque and speed.

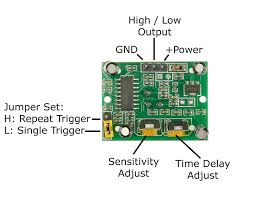
1. Easy Driver stepper motor board



This board is a great help for synchronizing and timing the phase changes of the stepper motor. This board has a precompiled code to move the motor in both directions according the the command given. This has 3 input pins: Direction, movement and GND. Whenever the movement pin receives a digital high signal it move to motor once in a 360 degree turn. It also determines the direction depending if the direction pin received a digital high ('1') or digital low ('0') signal. The speed cannot be adjusted, with the help of this board, but for our application it is not needed. If the speed was a factor, it could be adjusted with the help of a PWN signal.

1. König CSBLUEKEY200

This is the Bluetooth dongle used on the Raspberry side of the communication. Raspberry PI 3 has a built in bluetooth module, but our version does not include this. Without it the whole system would be paralyzed.

1. HC-SR501

The HC-SR501 is a PIR sensor working on between 4.5-20V. If motion is detected by the infrared sensor a logical ‘high’ is sent the its output which is transmitted to the microcontroller used (in this case an Arduino Mega). The pir sensor has to knobs. One adjusts the length a sensitivity (up to 7m) and the other the delay until a digital ‘low’ is sent on the output. This can be adjusted between (0.3s-5min). Another feature of the sensor is a jumper which sets to sensor to single or repeated triggering mode. In single trigger mode, if motion is detected once, the output is set to logical ‘high’ for the duration set on the knob, and then a logical ‘low’ is sent and motion between the two events are ignored. In repeat trigger mode the output is set to logical ‘high’ for the duration of the motion + the delay time.

Software

The Code was written in C language using the default arduino libraries. The used libraries included the serial communication and DateTime libraries. Unfortunately the timing of the motors is not automated at the final presentation of the project. The concept is to ask for a correct time whenever the arduino is reset and this synchronizes the internal clock of the arduino. After this the time could be defined for a morning motor control up and an evening control down time. For the morningAlarm() (motor up), when the given time has come +/-30 minutes, because of the randomization, a logical ‘high’ signal will be sent to the I/O pin which the motor control step is connected to, the amount of times needed for the motor to descend the shades. One rising edge signal results in a 360° turn of the stepper motor). Another logical ‘high’ signal will be sent to the I/O pin of the arduino board which the direction pin of the motor controller is connected to. For the eveningAlarm() (motor down), the randomization is the same, and the step impulses as well, but a logical ‘low’ is sent to the dir pin of the motor controller.

The randomisation uses the random library of the arduino. The randomSeed() is set to an unconnected analog pin of the arduino and the random noise received on the given analog input pin will set the seed to a given value each time the sketch is run. The random function will have a maximum value of 60, minimum is default which is 0. So the values returned will be between 0-59. If the given number is between 0-29 the time will be set to the alarm time - the returned value of the random function. If the value is between 29-59 the alarm time will be given time + (60-random seed returned value). The value of the time is in minutes. After the motor movement has finished, the arduino will ask for another clock sync. The Clock of the CU is synchronized over the internet. This can be set by the root user of the Raspbian OS at installation.

The motors can also be controlled manually. Manual control overwrites the alarms, and will be disabled until set again. Manual control works with the use of the keyboard of the CU. If the letter ‘f’ (lowercase) is pushed, the motor will operate in the same way as the morningAlarm(). If the letter ‘l’ (lowercase) is pushed, the motor will imitate the behavior of the eveningAlarm(). The user will also receive feedback on the serial monitor: “motor moving down”, “motor moving up”, “movement done”.The software also has limited safety and error handling implemented. If the motor is in any final state (up or down) then the motors cannot be moved in that direction further. If this is from a manual overwrite an error message will be sent to the CU via bluetooth and will be displayed on the serial monitor. During operation of the motor the giving input is disabled and ignored. The order will not be stored or remembered in any way and executed after the movement of the motors is finished. Unfortunately the only way to stop the motors is by removing the power source of the motors. Future improvements will implement a manual stop overwrite.

The software implementation of the pir sensor is also done in C with the default Arduino libraries. If the PIR sensor detects motion it will send a logical ‘high’ to the Arduino which will trigger a buzzer. This buzzer is on a 3 second timer at the moment, but future implementation will need a manual acknowledgement on the CU. The range and delay times of the pir sensor have mean described in the hardware section under HC-SR501.

# Verification tests

I will start with the easiest verification test first.

The sound test: The CU will have peripheries linked to it and running a Raspberry Linux OS (Raspbian). From here a sound file will be started, that the end product will use on a multimedia application, and test if the speakers and sound clip work(alarm.mp3).

The motion sensors test: visual and audio tests. First the visual test. A large object will be placed in front of the sensors and if they are activated a LED will flash with a 1 second interval on the CU.

Next the sound test. The same object will be placed in front of the sensors and instead of the LED the speakers will indicate,with the start of the alarm.mp3 file, that the motion sensors are operational and functional.

Additional quality checks can be done via placing smaller objects in front of the sensors and from various angles.

The motors and the timer test. These shall be done together, various times will be set on the timer. The internal clock of the CU will be set to to 31 minutes before the set time. If the timer was correctly implemented the motors will be operational within the hour. For better automation the motion sensors will be placed in front of the shades and when the motors are activated, a loud beeping will indicate this.

**Errors along to way**

The final version has very little to do with the original design. I have learned many new experiences and if I could start over I would do the planning and research phases differently. Most of the elements have changed at least once, and I have had to order several extras to make such at least one works. One of the main changes is the fact that there is no BLE module in this version. At first I wanted to use a BLE module to reduce the number of components used in my design, but the module I found appropriate, with the correct number of digital inputs and a chip antenna, didn't have a free SDK or even a trial version or student discount. I was not willing to pay $300 for this SDK and wait 2 weeks for my activation code to arrive. This was due to the fact that I was a greenhorn in the matter and didn't do a thorough even investigation into the module. I was only checking one step at a time and not thinking multiple steps ahead. This was mainly due to the fact I have never done a such a complex project before. Mainly just hobby “garage” projects for my own curiosity. For the new module I tried to find one, with not just the correct physical parametres, but also one that could be programmed using the arduino IDE. This was the most widely supported IDE that had linux support as well. This was needed as I prefer using this operating system and also the fact that the Raspberry PI runs a raspberian OS which is UNIX based. An appropriate BLE module was located, but as I was running out of time the order would not have arrived in time. Luckily I managed to locate an Arduino Mega 2560, same as the Mega just has an improved design. This was not the most efficient solution, but the best under the circumstances. Apart from this, as most of my components was ordered from china, not all were up to standard. Luckily I had some experience with ordering from aboard and with the low cost per item, I ordered spares. The project was a great learning experience for me, and I think I have learned a lot from my own mistakes which could be applied for the future improvements I will make to this system.

**Future Improvements**

* Mobile App
* More utilities online
* Increased Security
* Time sync from web and sunrise/sunset times
* Stored info (Energy Usage, App login times)

**Implementation**

This system could be used in any home or office which had valuables stored inside which would be wanted to kept safe. Garages and storage units could also use this concept. With the future improvements made, it would also be used as a device to measure power usage and could be implemented in factories and other buildings were power was not needed 24/7. Server rooms could also find this technology useful as the hardware is already available, and many extra useful information could be read out. Also any home which wanted more automation could implement this technology. Another positive is that it uses a 50m range bluetooth signal, which will not use the families internet bandwidth.

Varga Barna

HKH03C