

**DEPARTMENT OF**

**ELECTRICAL AND ELECTRONICS ENGINEERING**

**EEM482**

**Controlling an Arduino Board with**

**Personal Computer over a Server**

**Sekran Mert KILIÇ 52522588108**

[**sekranmertkilic@gmail.com**](mailto:sekranmertkilic@gmail.com)

**Muhammed Safa ŞAHİNER 65797251844**

[**muhammedsafasahiner@gmail.com**](mailto:muhammedsafasahiner@gmail.com)

**Supervisor: Assist. Prof. Dr. İsmail San**

[**isan@eskisehir.edu.tr**](mailto:isan@eskisehir.edu.tr)

AWS-Windows

# **Abstract**

Today, in lots of environment IOT systems are widely used. Smart home systems are used in a wide range of applications from the IOT industry to daily usage, all around the world for this purpose. This project is to achieve a relatively cheap and easy to implement IOT smart home system. Server to client model is used for this system. For server AWS’ EC2 instance is used. An Arduino board is used for smart house system controller and to receive commands from server. In this system, arduinos and PCs are the clients. Python language and TCP protocol is used to implement the system.

# **Introduction**

IOT systems, with newly developed smart house technology are a part of our daily lives. To develop a system that is cheap and easy to modify, Arduinos, PCs and AWS instances are used. This system proposes a server to client model to achieve communication between arduinos and PCs. Tcp protocol is used, because with Tcp a socket is created between a client and server. In this project, server separates its clients to two types arduinos and clients. These types are saved as object lists with their addresses, names, and sockets on server. With this lists its easy to select between arduinos or clients.

# **Server**

Server runs on a windows server on AWS. A windows server is used because of its easy-to-use graphic interface. For server program, python language used. As libraries, socket library for network communication, threading library for client threads and datetime library for server time are used. Socket library has a function, “.socket” which creates a socket with private IP and port. .AF\_INET and .SOCKSTREAM protocols are used to create a TCP socket. On server program, a main loop which accepts clients, saves their addresses, names, and types on a class list, and starts a thread to continue communication is written. This loop asks type and names with one-byte messages(“t,n”). One-byte messages are used because of arduinos limited memory. The socket object class which has a client’s name, address and socket object is used for both arduinos and clients list.

A receiveMessage function which handles the messages received from clients is written. This function runs as a thread with thread library. Message received from client is separated to its tokens by this function. First token implements the command. First token can be: “-s (send), -t (time), -l (list), -q (quit)”. -s command can send messages to either an Arduino nor a client. For messages that sent to a client, their first three tokens are deleted. For messages that sent to arduinos, their 5th token is used to select a one-byte command. For commands that sent to arduinos a message that declares the command is broadcasted to all clients on the clients list. -t command returns a time string to the client using datetime library. -l command creates a string using the clients list and the arduinos list and returns it to client. -q command removes client from list and close its connection with server. A sendClient function is written to send a message to client. This function uses an for loop that searchs client list and send message to receiver. For messages which sent to arduinos a sendArduino function is written. This function uses 5th token of the message and sends a one-byte message(ascii “a” for led on, ascii ”b” for led off, ascii “c” for led status) to Arduino. To broadcast commands a broadcastMessage function is written. This function uses a for loop in the clients list and sends declaration message to every client on the list.

# **Client**

Python language used for client program too. With socket programs .socket function, using server’s public IP and port number a Tcp socket created. Client program has two function one to receive and one to send messages. recvTh function is used to print messages that received from server. This function replies “t” message which asks for client type with “c”(client) and asks users name for “n” message and replies with user input. Other than these two, it prints other messages to terminal. Sendth function sent messages to server except “-h” which prints command list for user.

# **Arduino**

Arduino uses an ethernet shield (Wiznet W5100) for connecting to the internet. Ethernet.h library used. A client object is created. A mac address that can be found on the shield and an IP address which must be on lan is assigned to Arduino. With those and DNS, shield is connected to ethernet. With server IP and its port, the client object is connected to the server. In main loop, using client.available() function which returns the number of bytes available for reading, message received from server is written on a buffer. when this message is “t” Arduino replies with “a”(for arduino), when it is “n” Arduino replies with a name string, when message is “a” led that is connected to digital pin 8 is set high, when it is “b” led is set to low, and when it is “c” Arduino sends digital pin 9 value as a string which is connected to led.

# **Problems**

While programming the first problem that occurred was a connection problem. Because of windows OS just a few ports could be used by PyCharm. To solve this problem, by windows resources manager the Tcp port that used by PyCharm is found and enabled on EC2 instance settings. Second problem was the AWS system did not give a static IP, that problem was solved when it is realized that the IP on EC2 instance and the one on program were different. The third problem occurred on Arduino. Firstly, the message which received by Arduino was a string of command that must be processed by Arduino. This was not a preferred because of Arduino’s low memory. Then 3-byte messages are used to convey commands. But this method was not a preferable way, because a 3-byte buffer couldn't be processed. To solve this, one-byte messages are used. Because they were processed easy and fast.

# **Code**

Code can be found at <https://github.com/sekranmert/AWS-Arduino-SmartHomeSystem>