KOCAELİ UNIVERSITY FACULTY OF ENGINEERING

By Using the Joystick Module Reading the Analog Data from the X and Y Axes and Writing the Data to the Thingspeak Environment with Wifi Module

MICROCONTROLLER PROECT

Ahmet Haydar ERDEM / 140207064

Electronic and Communication Engineering

Advisor: Prof. Dr. Oğuzhan URHAN

KOCAELİ,2018

Table of Contents

1. INTRODUCTION

In today's technology has become commonplace and necessary to process high-dimensional data. This necessity has brought us some disadvantages such as storage capacity, high cost, low productivity. Cloud technology has been developed in order to prevent these. Having the ability to create very large data storage areas with the virtual storage areas in the web, cloud technology nowadays has a flexible structure with data transfers, common data usage, private and general usage. This allows the user to provide lower cost and faster data transfer capability.

In the project, it is aimed to send different data from the microcontroller using different channels to the cloud and to store the data.

2. COMPONENTS

2.1. Joystic Module

Image 1. Analog Module

It is a module that has analog outputs for X - Y axes. The analog values read from the X and Y axes are intended to be meaningful in the microcontroller and written in the cloud environment.

2.2. ESP8266 Wifi Module

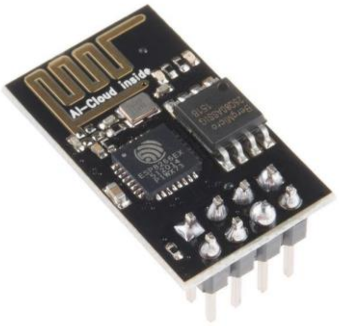
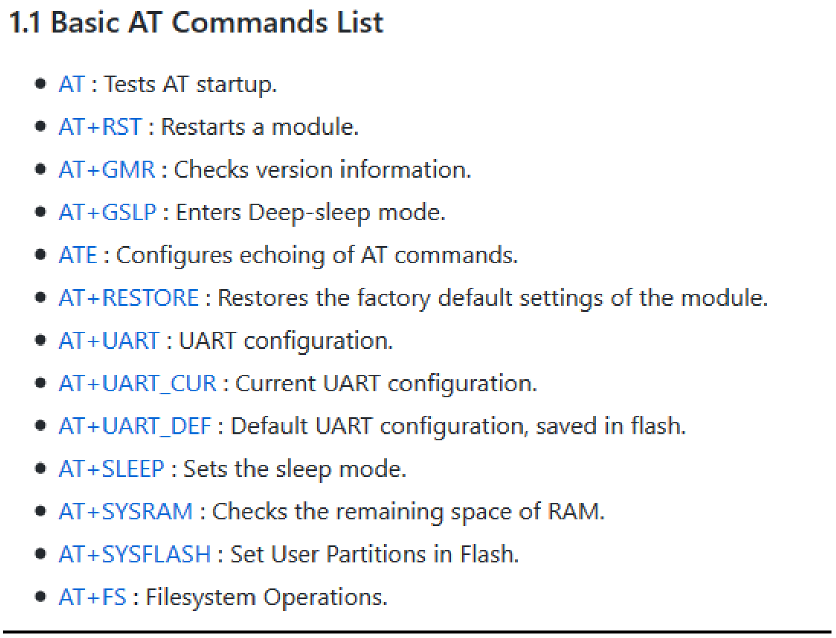


Image 2. ESP8266 Wifi Module

Before describing the Wifi module, it is necessary to talk about the concept of IoT (The Internet of Things). In short, IOT is a concept that aims to access internet each object and communicate with other devices. Smart bracelets, smart watches, smart glasses, smart cars, home automation systems that are popular nowadays are the products of IOT. The wifi module assumes the role of bridge between objects and the internet. In our project, we will write the data in the cloud that we will obtain by bringing accessable the microcontroller to the internet. Communication between the microcontroller and the wifi module is provided by the UART communication protocol. The microcontroller and wifi module have receiver and transmitter (TX - RX). Therefore, communication is made meaningful according to the sent commands and the received answers. Thus, the targeted operation is performed.

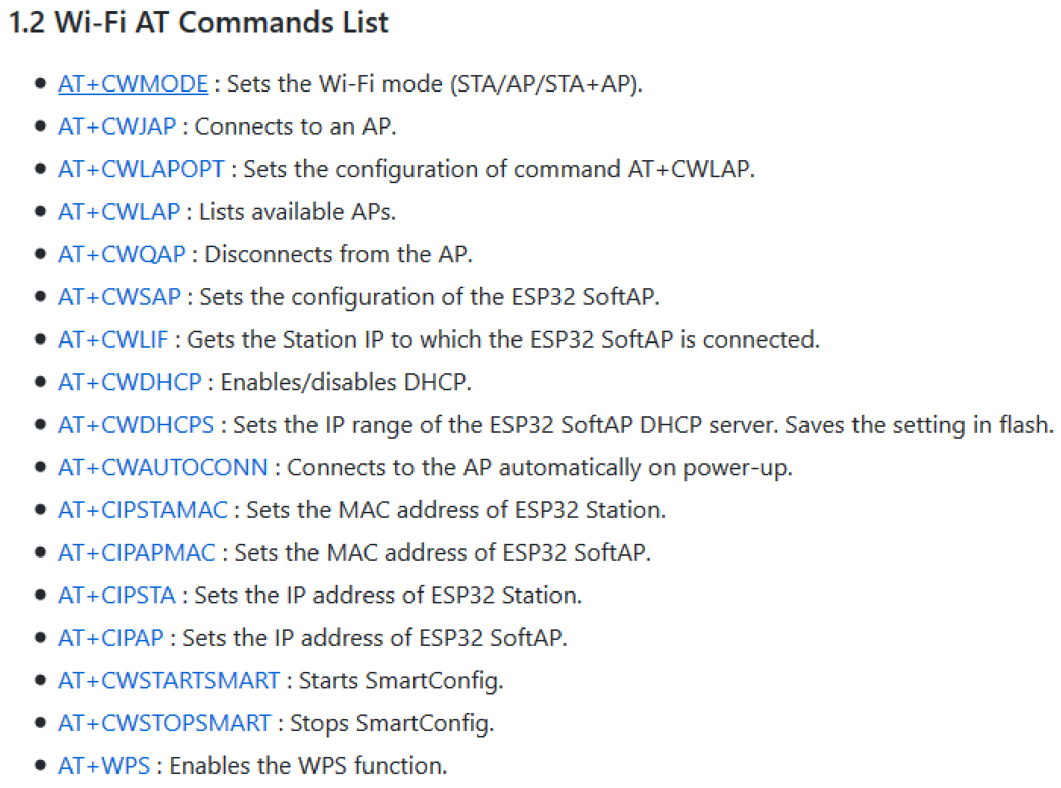
The operating environment of the ESP8266 module consists of AT commands. Communication is configured via commands and communication with the desired object is made.

2.2.1. AT Commands



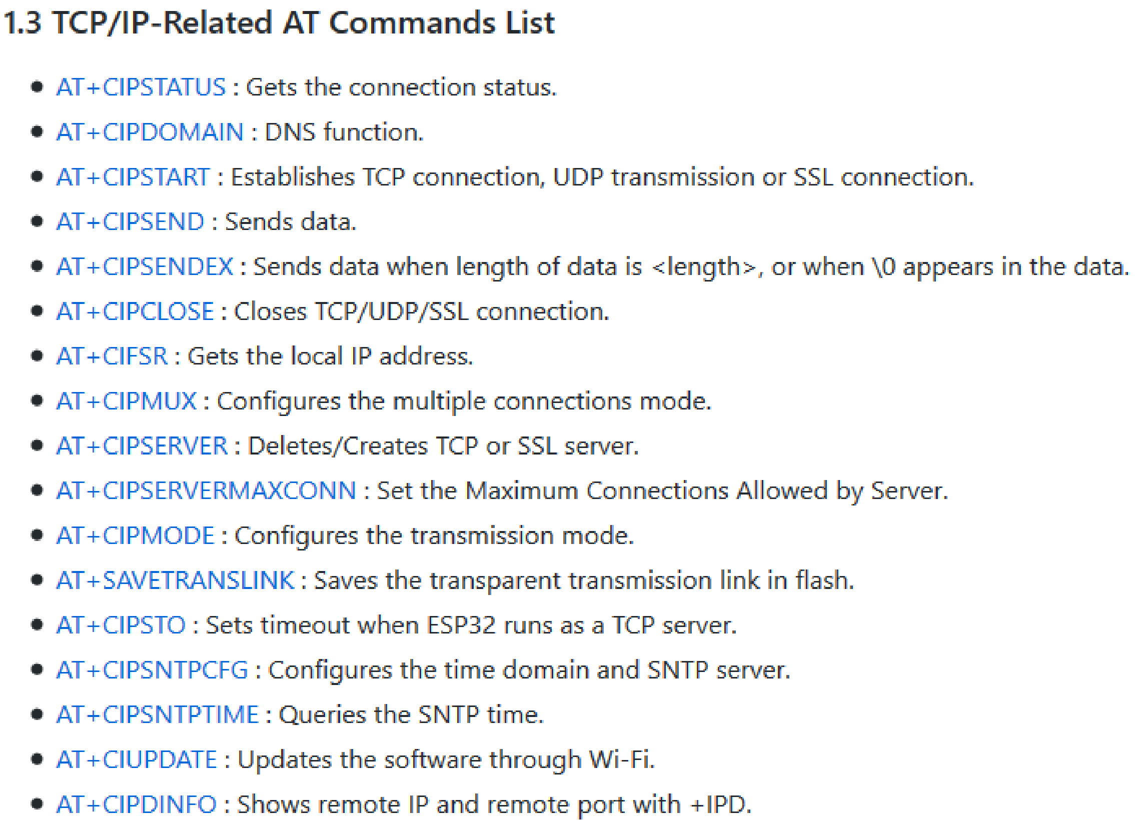
1.1 Basic AT Commands List

- AT : Tests AT startup.
- AT+RST : Restarts a module.
- AT+GMR : Checks version information.
- AT+GSLP : Enters Deep-sleep mode.
- ATE : Configures echoing of AT commands.
- AT+RESTORE : Restores the factory default settings of the module.
- AT+UART : UART configuration.
- AT+UART_CUR : Current UART configuration.
- AT+UART_DEF : Default UART configuration, saved in flash.
- AT+SLEEP : Sets the sleep mode.
- AT+SYSRAM : Checks the remaining space of RAM.
- AT+SYSFLASH : Set User Partitions in Flash.
- AT+FS : Filesystem Operations.



1.2 Wi-Fi AT Commands List

- AT+CWMODE : Sets the Wi-Fi mode (STA/AP/STA+AP).
- AT+CWJAP : Connects to an AP.
- AT+CWLAPOPT : Sets the configuration of command AT+CWLAP.
- AT+CWLAP : Lists available APs.
- AT+CWQAP : Disconnects from the AP.
- AT+CWSAP : Sets the configuration of the ESP32 SoftAP.
- AT+CWLIF : Gets the Station IP to which the ESP32 SoftAP is connected.
- AT+CWDHCP : Enables/disables DHCP.
- AT+CWDHCPS : Sets the IP range of the ESP32 SoftAP DHCP server. Saves the setting in flash.
- AT+CWAUTOCONN : Connects to the AP automatically on power-up.
- AT+CIPSTAMAC : Sets the MAC address of ESP32 Station.
- AT+CIPAPMAC : Sets the MAC address of ESP32 SoftAP.
- AT+CIPSTA : Sets the IP address of ESP32 Station.
- AT+CIPAP : Sets the IP address of ESP32 SoftAP.
- AT+CWSTARTSMART : Starts SmartConfig.
- AT+CWSTOPSMART : Stops SmartConfig.
- AT+WPS : Enables the WPS function.



1.3 TCP/IP-Related AT Commands List

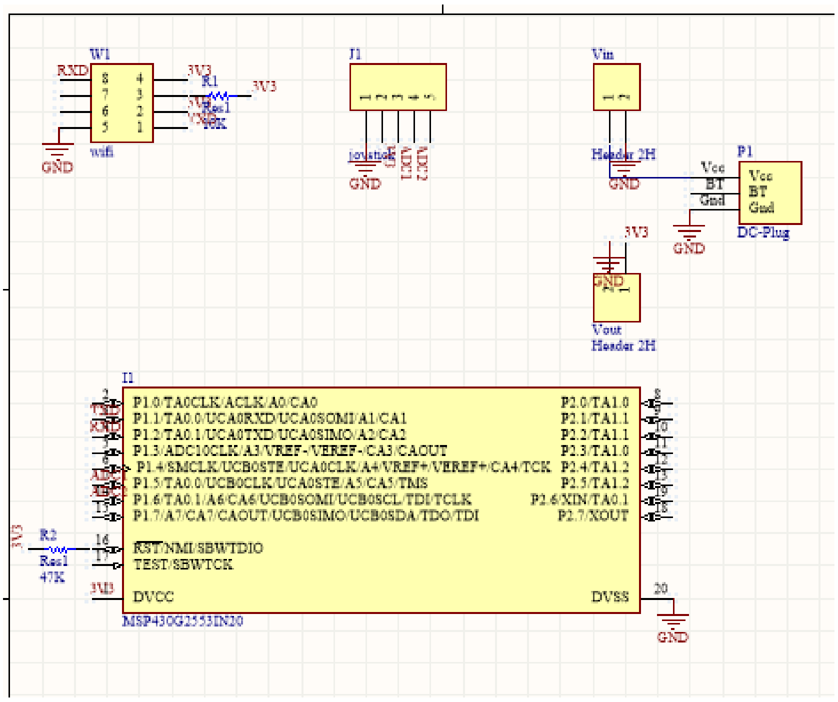
- AT+CIPSTATUS : Gets the connection status.
- AT+CIPDOMAIN : DNS function.
- AT+CIPSTART : Establishes TCP connection, UDP transmission or SSL connection.
- AT+CIPSEND : Sends data.
- AT+CIPSENDEX : Sends data when length of data is <length>, or when \0 appears in the data.
- AT+CIPCLOSE : Closes TCP/UDP/SSL connection.
- AT+CIFSR : Gets the local IP address.
- AT+CIPMUX : Configures the multiple connections mode.
- AT+CIPSERVER : Deletes/Creates TCP or SSL server.
- AT+CIPSERVERMAXCONN : Set the Maximum Connections Allowed by Server.
- AT+CIPMODE : Configures the transmission mode.
- AT+SAVETRANSLINK : Saves the transparent transmission link in flash.
- AT+CIPSTO : Sets timeout when ESP32 runs as a TCP server.
- AT+CIPSNTPCFG : Configures the time domain and SNTP server.
- AT+CIPSNTPTIME : Queries the SNTP time.
- AT+CIUPDATE : Updates the software through Wi-Fi.
- AT+CIPDINFO : Shows remote IP and remote port with +IPD.

2.3. Other Components

The connector for the external power circuit and external power connection is used for portable operation of the design we designed.

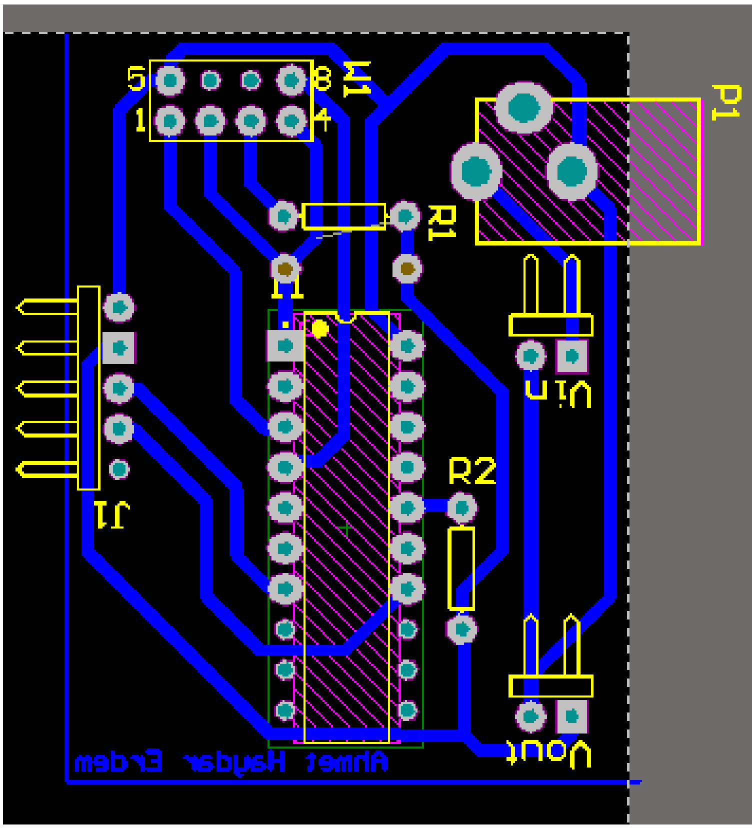
3. ALTIUM DESIGNS

3.1. Schematic Design



In the schematic drawing, labels were used for a simpler look. The receiver and transmitter pins of the wifi module and microcontroller are connected in reverse order for the communication to occur.

3.2. PCB Design



The design of the schematic drawing was overwritten on the PCB file. So the PCB was designed to fit the schematic diagram.

4. REFERENCES

http://www.mcu-turkey.com/msp430-launchpad-msp430g2553-donanimsal-uart-kullanimi/

https://github.com/espressif/esp32-at/blob/master/docs/ESP32_AT_Commands_Set.md#cmd-STARTS

https://github.com/espressif/esptool/wiki/ESP32-Boot-Mode-Selection

https://www.embeddedrelated.com/showarticle/420.php

http://www.simplyembedded.org/tutorials/msp430-uart/

https://github.com/amartinezacosta/MSP432-ESP8266/blob/master/Devices/ESP8266.c

https://room-15.github.io/blog/2015/03/26/esp8266-at-command-reference/