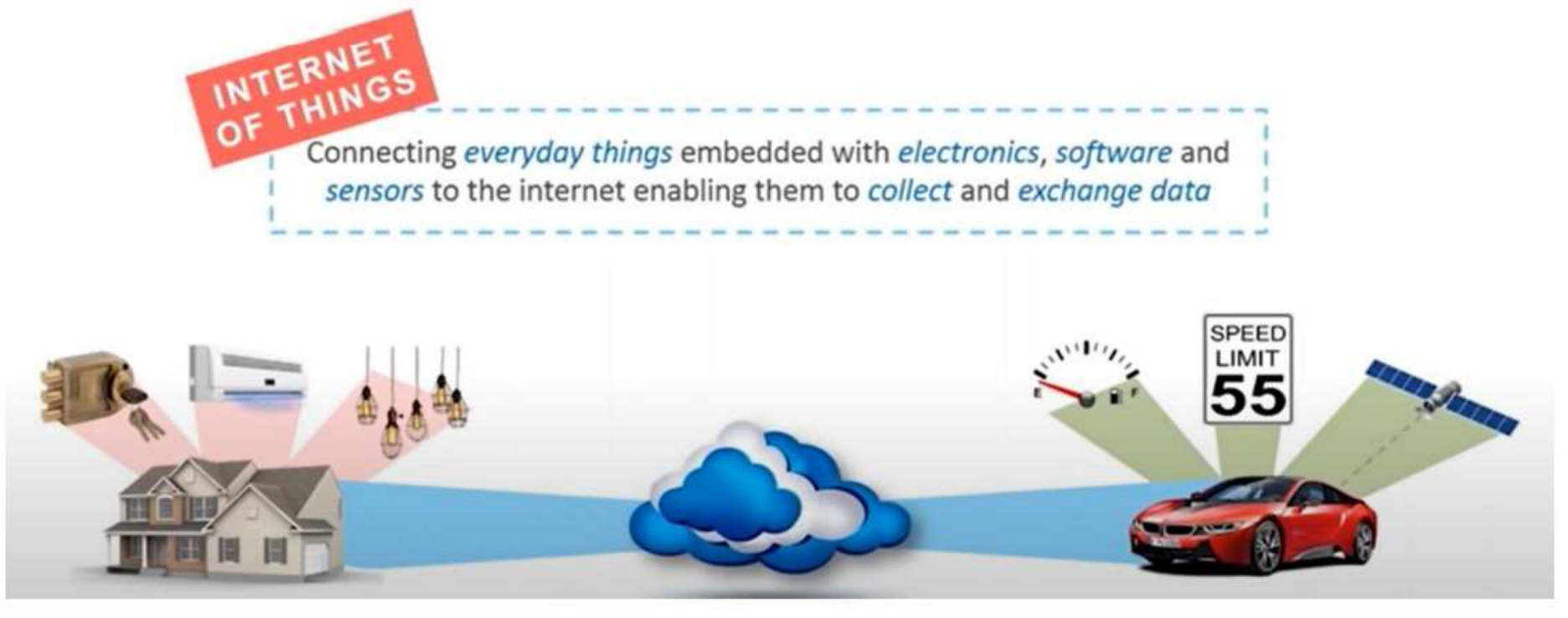

LIVE
LEARN
EMBEDDED SYSTEM
ARM CORTEX M4
DAY 29/30
(30 DAYS CHALLENGE)
M K JEEVARAJAN

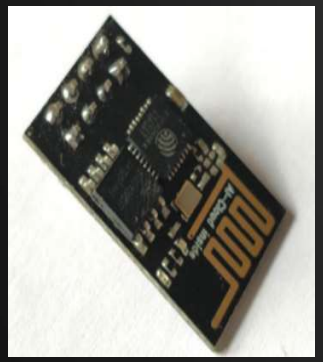
What you will Learn Today?

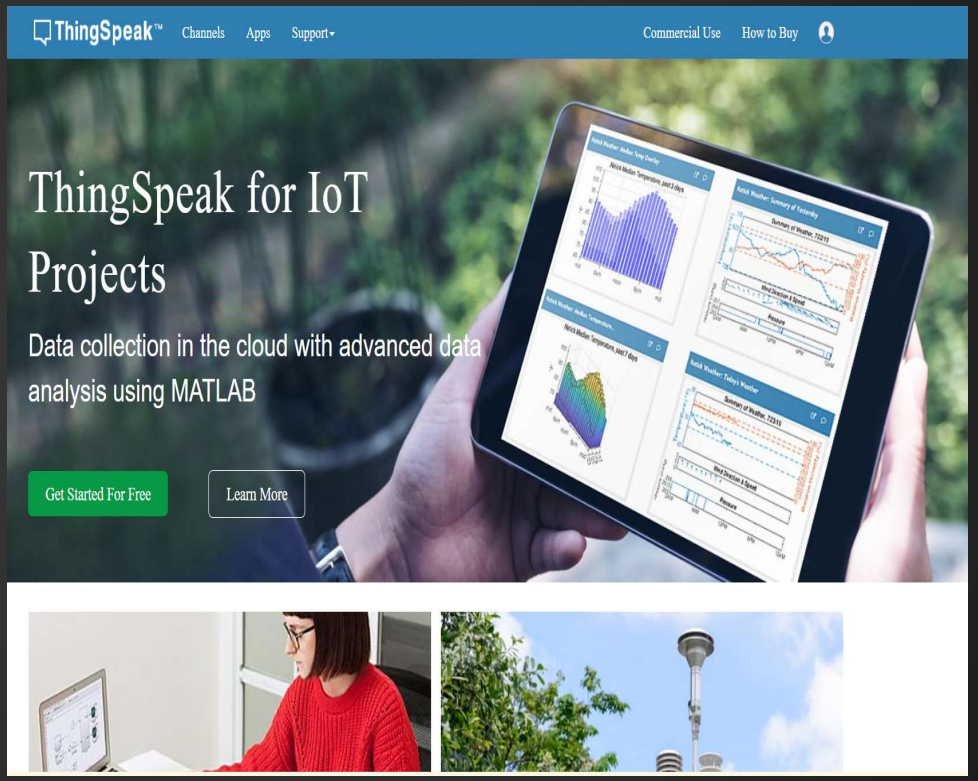
- ✓ IoT Based Data Monitoring System using ARM CORTEX M4 & ESP8266 –Thingspeak Cloud

What is Internet of Things



Hardware Required



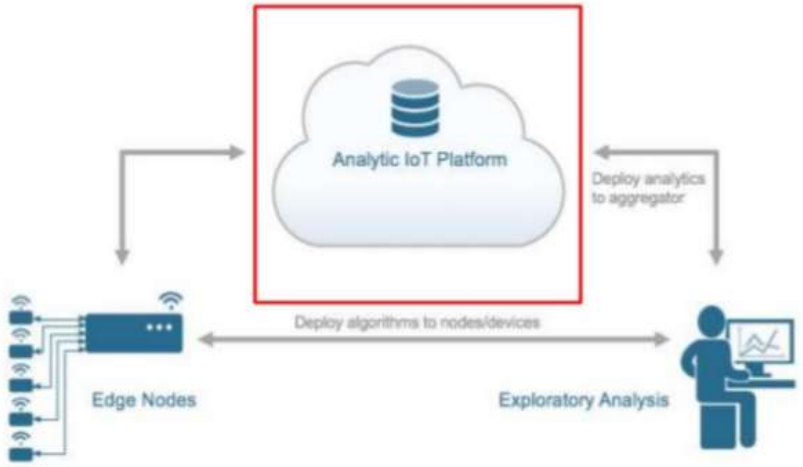


SOFTWARE REQUIRED

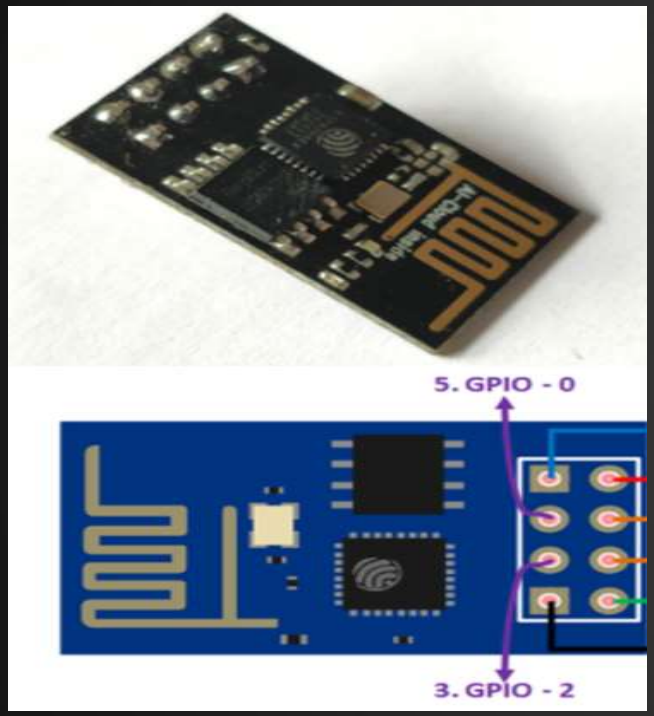
- STM2CUBE IDE
- SALEA LOGIC ANALYZER
- Thingspeak

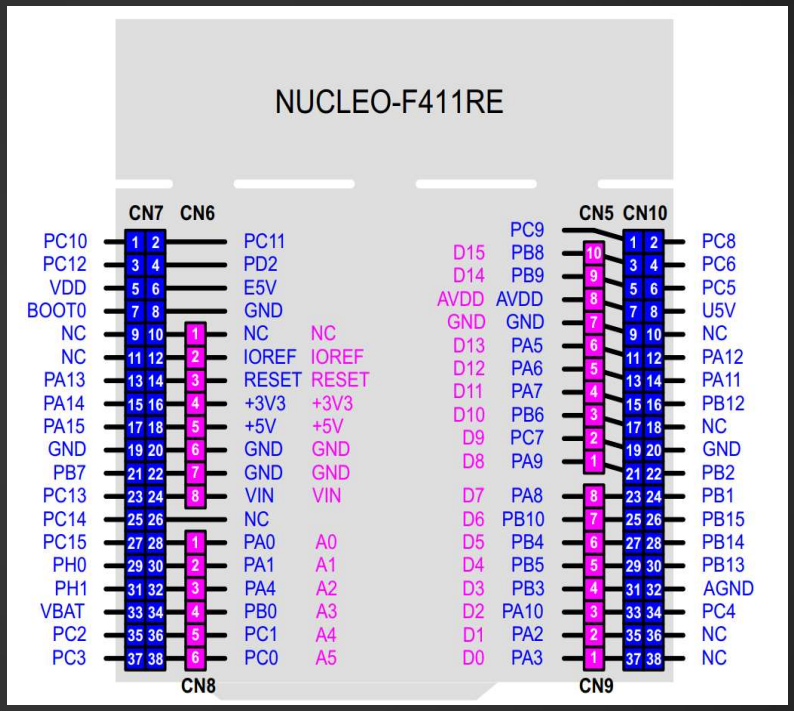
What is Thingspeak

- Analytic IoT platform
  - Collect data from sensors, "things"
  - Visualize data instantly
  - Has more than 60,000 users
- Analyze data
  - MATLAB integration allows users to run scheduled code on data coming into ThingSpeak
- Act on data
  - E.g. send a tweet when the temperature in your backyard reaches 32 degrees



SCHEMATIC DESIGN



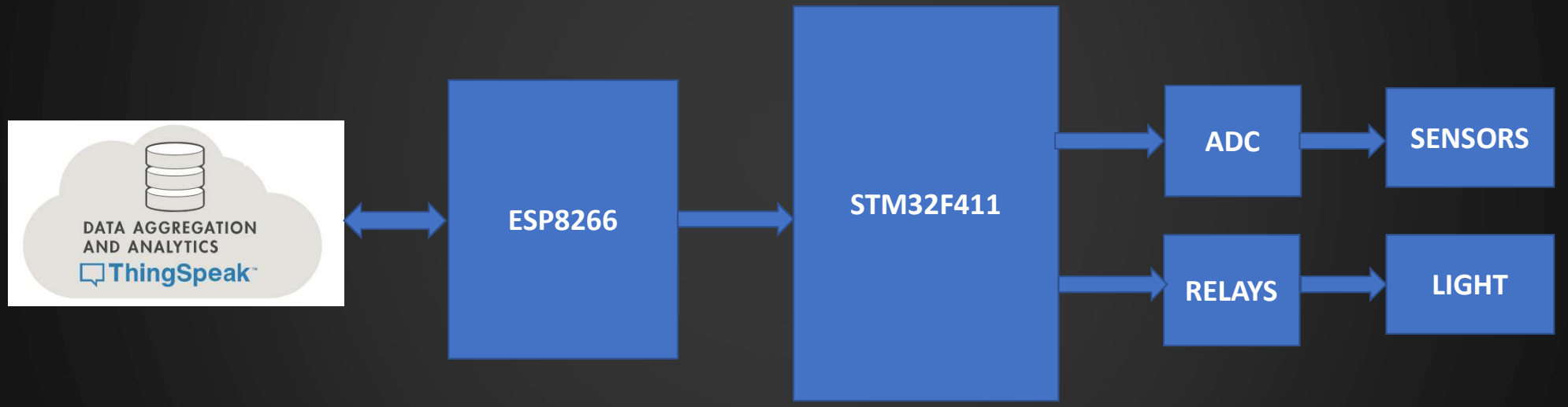


//PB6->TX

//PB7->RX ->ESP8266 TX

//RELAY -PA5

Block Diagram



UART Parameters

- Baud Rate 115200
- Number of Data Bits (7, 8)
- Parity Bit (On, Off)
- Stop Bits (0, 1, 2)
- Flow Control (None, On, Hardware)

AT COMMANDS

AT commands used for data transmission

- AT+RST
- AT+CWJAP="WIFINAME", "Password"
- AT+CIPSTART="TCP","184.106.153.149",80
- AT+CIPSEND=49
- GET /update?api_key=XXXXXXXXXXXXXXX&field1=000
- AT+CLOSE

Connection Details

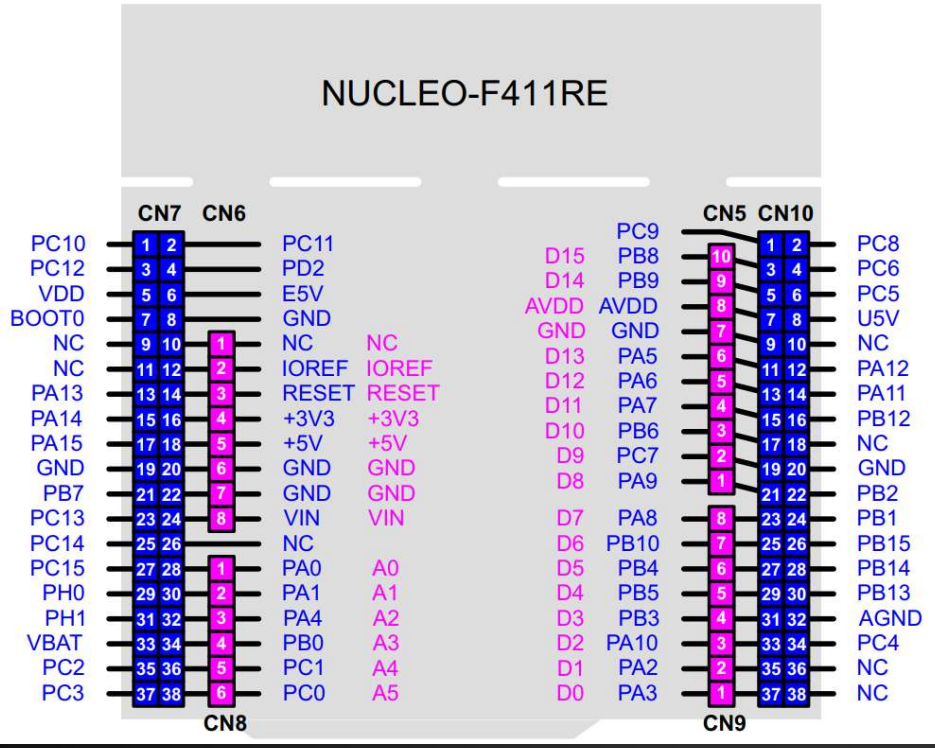


Table 9. Alternate function mapping (continued)

Port		AF00	AF01	AF02	AF03	AF04	AF05	AF06	AF07	AF08	AF09
		SYS_AF	TIM1/TIM2	TIM3/ TIM4/ TIM5	TIM9/ TIM10/ TIM11	I2C1/I2C2/ I2C3	SPI1/I2S1S PI2/ I2S2/SPI3/ I2S3	SPI2/I2S2/ SPI3/ I2S3/SPI4/ I2S4/SPI5/ I2S5	SPI3/I2S3/ USART1/ USART2	USART6	I2C2/ I2C3
Port B	PB0	-	TIM1_CH2N	TIM3_CH3	-	-	-	SPI5_SCK /I2S5_CK		-	-
	PB1	-	TIM1_CH3N	TIM3_CH4	-	-	-	SPI5_NSS /I2S5_WS		-	-
	PB2	-	-	-	-	-	-	-	-	-	-
	PB3	JTDO-SWO	TIM2_CH2	-	-	-	SPI1_SCK/I 2S1_CK	SPI3_SCK /I2S3_CK	USART1_ RX	-	I2C2_SDA
	PB4	JTRST		TIM3_CH1	-	-	SPI1_MISO	SPI3_MISO	I2S3ext_S D	-	I2C3_SDA
	PB5	-	-	TIM3_CH2	-	I2C1_SMB A	SPI1_MOSI /I2S1_SD	SPI3_MOSI/ I2S3_SD		-	-
	PB6	-	-	TIM4_CH1	-	I2C1_SCL	-	-	USART1_ TX	-	-
	PB7	-	-	TIM4_CH2	-	I2C1_SDA	-	-	USART1_ RX	-	-
	PB8	-	-	TIM4_CH3	TIM10_CH1	I2C1_SCL	-	SPI5_MOSI/ I2S5_SD	-	-	I2C3_SDA
	PB9	-	-	TIM4_CH4	TIM11_CH1	I2C1_SDA	SPI2_NSS/I 2S2_WS	-	-	-	I2C2_SDA

STEPS FOR UART TRANSMIT

//Enable clocK access to GPIOA

//SET PA2,PA3 MODE TO Alternate function Mode

//SET PA2 ,PA3 Alternative function type to UART_TX (AF07)

//CONFIGURE UART MODULE

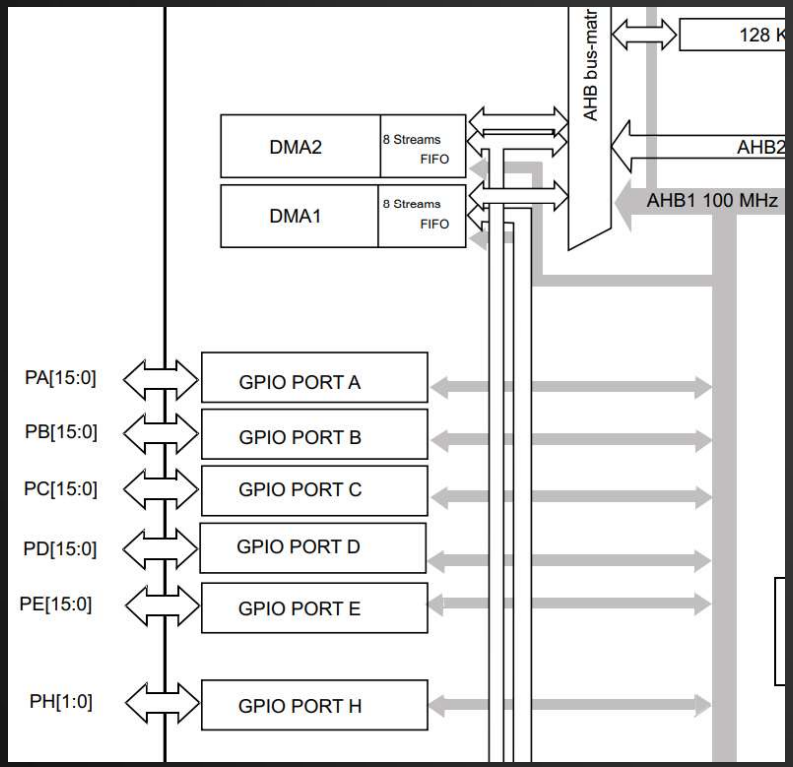
//ENABLE CLOCK ACCESS TO UART2

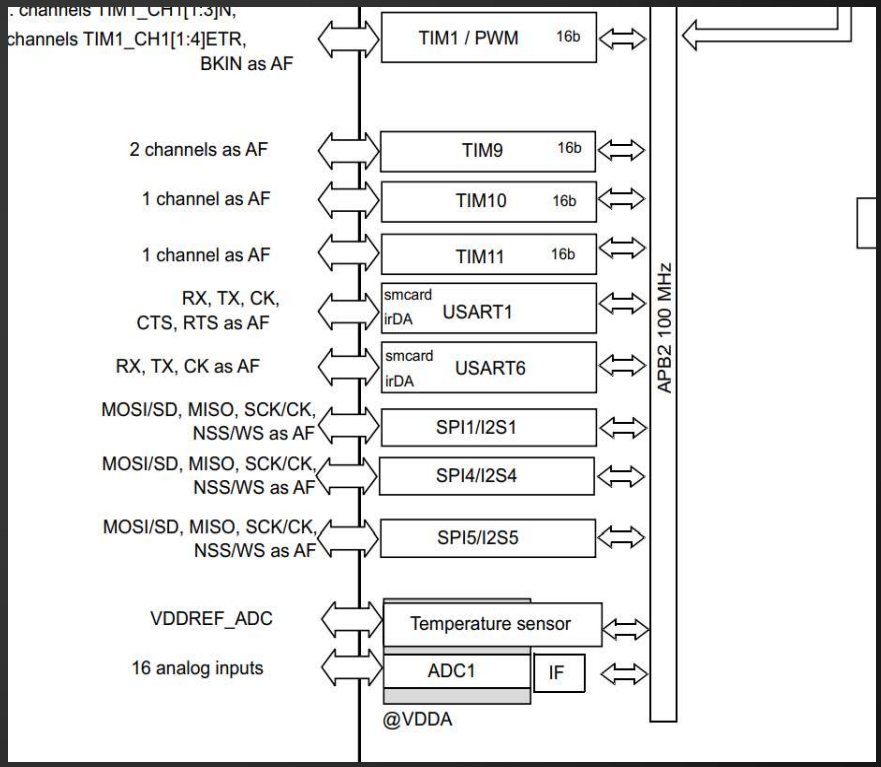
//CONFIGURE BAUDRATE

//CONFIGURE TRANSFER DIRECTION

//ENABLE UART MODULE

Block Diagram





USART1-FUNCTIONAL MAPPING

Port		AF00	AF01	AF02	AF03	AF04	AF05	AF06	AF07	A
		SYS_AF	TIM1/TIM2	TIM3/ TIM4/ TIM5	TIM9/ TIM10/ TIM11	I2C1/I2C2/ I2C3	SPI1/I2S1SPI2/ I2S2/SPI3/ I2S3	SPI2/I2S2/ SPI3/ I2S3/SPI4/ I2S4/SPI5/ I2S5	SPI3/I2S3/ USART1/ USART2	US
Port B	PB0	-	TIM1_CH2N	TIM3_CH3	-	-	-	SPI5_SCK /I2S5_CK		
	PB1	-	TIM1_CH3N	TIM3_CH4	-	-	-	SPI5_NSS /I2S5_WS		
	PB2	-	-	-	-	-	-	-	-	
	PB3	JTDO-SWO	TIM2_CH2	-	-	-	SPI1_SCK/I2S1_CK	SPI3_SCK /I2S3_CK	USART1_RX	
	PB4	JTRST		TIM3_CH1	-	-	SPI1_MISO	SPI3_MISO	I2S3ext_SD	
	PB5	-		TIM3_CH2	-	I2C1_SMBA	SPI1_MOSI /I2S1_SD	SPI3_MOSI/ I2S3_SD		
	PB6	-	-	TIM4_CH1	-	I2C1_SCL	-	-	USART1_TX	
	PB7	-	-	TIM4_CH2	-	I2C1_SDA	-	-	USART1_RX	

FUNCTIONAL MAPPING

Table 9. Alternate function mapping

Port		AF00	AF01	AF02	AF03	AF04	AF05	AF06	AF07	AF08	AF09	AF10	AF11	AF12	AF13	AF14	AF15
		SYS_AF	TIM1/TIM2	TIM3/ TIM4/ TIM5	TIM9/ TIM10/ TIM11	I2C1/I2C2/ I2C3	SPI1/I2S1SPI2/ I2S2/SPI3/ I2S3	SPI2/I2S2/ SPI3/ I2S3/SPI4/ I2S4/SPI5/ I2S5	SPI3/I2S3/ USART1/ USART2	USART6	I2C2/ I2C3	OTG1_FS		SDIO			
Port A	PA0	-	TIM2_CH1/ TIM2_ETR	TIM5_CH1	-	-	-	-	USART2_ CTS	-	-	-	-	-	-	-	EVENT OUT
	PA1	-	TIM2_CH2	TIM5_CH2	-	-	SPI4_MOSI /I2S4_SD	-	USART2_ RTS	-	-	-	-	-	-	-	EVENT OUT
	PA2	-	TIM2_CH3	TIM5_CH3	TIM9_CH1	-	I2S2_CKIN	-	USART2_ TX	-	-	-	-	-	-	-	EVENT OUT
	PA3	-	TIM2_CH4	TIM5_CH4	TIM9_CH2	-	I2S2_MCK	-	USART2_ RX	-	-	-	-	-	-	-	EVENT OUT
	PA4	-	-	-	-	-	SPI1_NSS/I 2S1_WS	SPI3_NSS/I2 S3_WS	USART2_ CK	-	-	-	-	-	-	-	EVENT OUT
	PA5	-	TIM2_CH1/ TIM2_ETR	-	-	-	SPI1_SCK/I 2S1_CK	-	-	-	-	-	-	-	-	-	EVENT OUT
	PA6	-	TIM1_BKIN	TIM3_CH1	-	-	SPI1_MISO	I2S2_MCK	-	-	-	-	-	SDIO_ CMD	-	-	EVENT OUT
	PA7	-	TIM1_CH1N	TIM3_CH2	-	-	SPI1_MOSI /I2S1_SD	-	-	-	-	-	-	-	-	-	EVENT OUT
	PA8	MCO_1	TIM1_CH1	-	-	I2C3_ SCL	-	-	USART1_ CK	-	-	USB_FS_ SOF	-	SDIO_ D1	-	-	EVENT OUT
	PA9	-	TIM1_CH2	-	-	I2C3_ SMBA	-	-	USART1_ TX	-	-	USB_FS_ VBUS	-	SDIO_ D2	-	-	EVENT OUT

ALTERNATE FUNCTIONS

Address offset: 0x20
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFRL7[3:0]				AFRL6[3:0]				AFRL5[3:0]				AFRL4[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFRL3[3:0]				AFRL2[3:0]				AFRL1[3:0]				AFRL0[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
				0	1	1	1								

Bits 31:0 **AFRLy:** Alternate function selection for port x bit y (y = 0..7)
These bits are written by software to configure alternate function I/Os

AFRLy selection:

0000: AF0	1000: AF8
0001: AF1	1001: AF9
0010: AF2	1010: AF10
0011: AF3	1011: AF11
0100: AF4	1100: AF12
0101: AF5	1101: AF13
0110: AF6	1110: AF14
0111: AF7	1111: AF15

BAUD RATE CALCULATION

- Peripheral Clock =16000000
- BAUDRATE=9600
- BRR =((16000000+(9600/2))/9600)
- BRR= 1667
- HEX VALUE OF BRR =0X0683

- BAUDRATE=115200
- BRR =((16000000+(115200/2))/115200)
- BRR= 139
- HEX VALUE OF BRR =0X008B

UART WRITE

```
void uart2_write(int ch)
{
    //Make sure the transmit data register is empty
    while(!(*USART2_SR & 0x0080)){}
    //write to transmit data register
    *USART2_DR =(ch&0XFF);
}
```

TXE: Transmit data register empty

This bit is set by hardware when the content of the TDR register has been transferred into the shift register. An interrupt is generated if the TXEIE bit =1 in the USART_CR1 register. It is cleared by a write to the USART_DR register.

0: Data is not transferred to the shift register

1: Data is transferred to the shift register)

26.6.1 Status register (USART_SR)

Address offset: 0x00

Reset value: 0x00C0 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CTS	LBD	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
						rc_w0	rc_w0	r	rc_w0	rc_w0	r	r	r	r	r

USART DATA REGISTER

Data register (USART_DR)

Address offset: 0x04

Reset value: 0xXXXX XXXX

Bits 31:9 Reserved, must be kept at reset value

Bits 8:0 **DR[8:0]**: Data value

Contains the Received or Transmitted data character, depending on whether it is read from or written to.

The Data register performs a double function (read and write) since it is composed of two registers, one for transmission (TDR) and one for reception (RDR)

The TDR register provides the parallel interface between the internal bus and the output shift register (see Figure 1).

The RDR register provides the parallel interface between the input shift register and the internal bus.

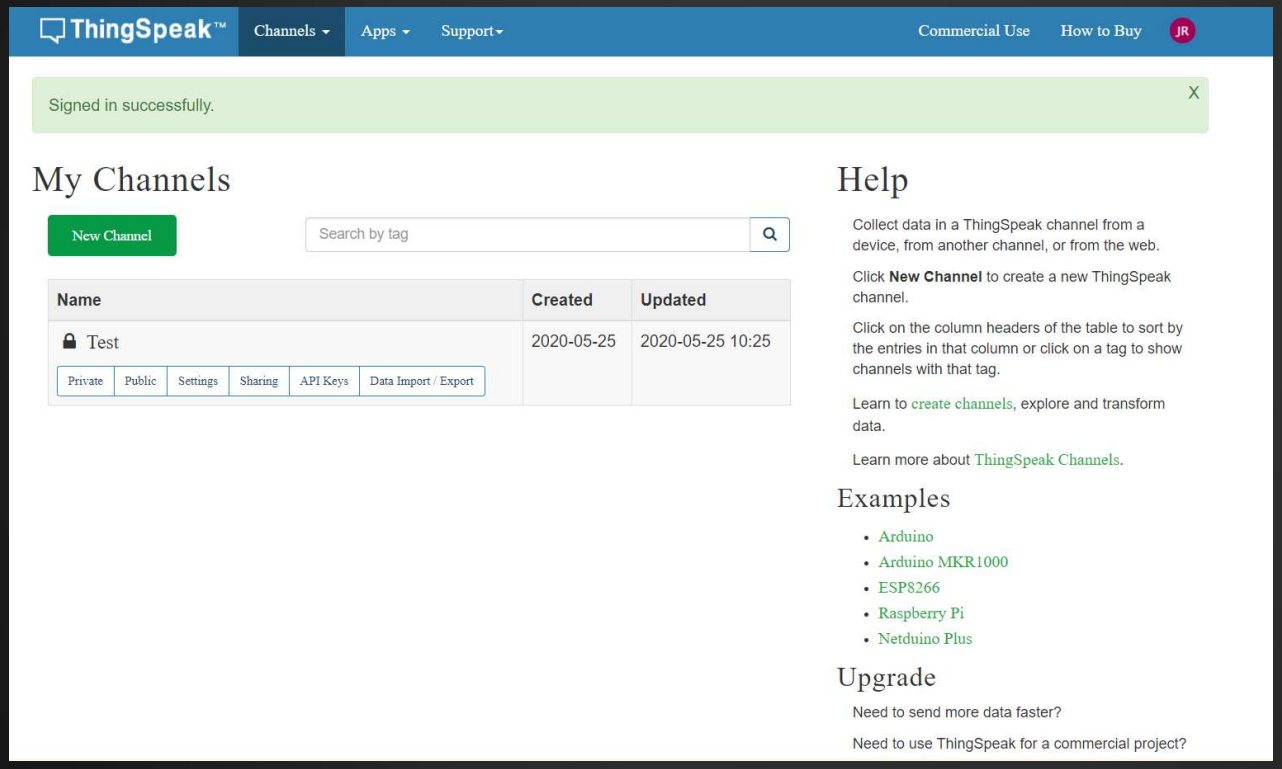
When transmitting with the parity enabled (PCE bit set to 1 in the USART_CR1 register), the value written in the MSB (bit 7 or bit 8 depending on the data length) has no effect because it is replaced by the parity.

When receiving with the parity enabled, the value read in the MSB bit is the received parity bit.

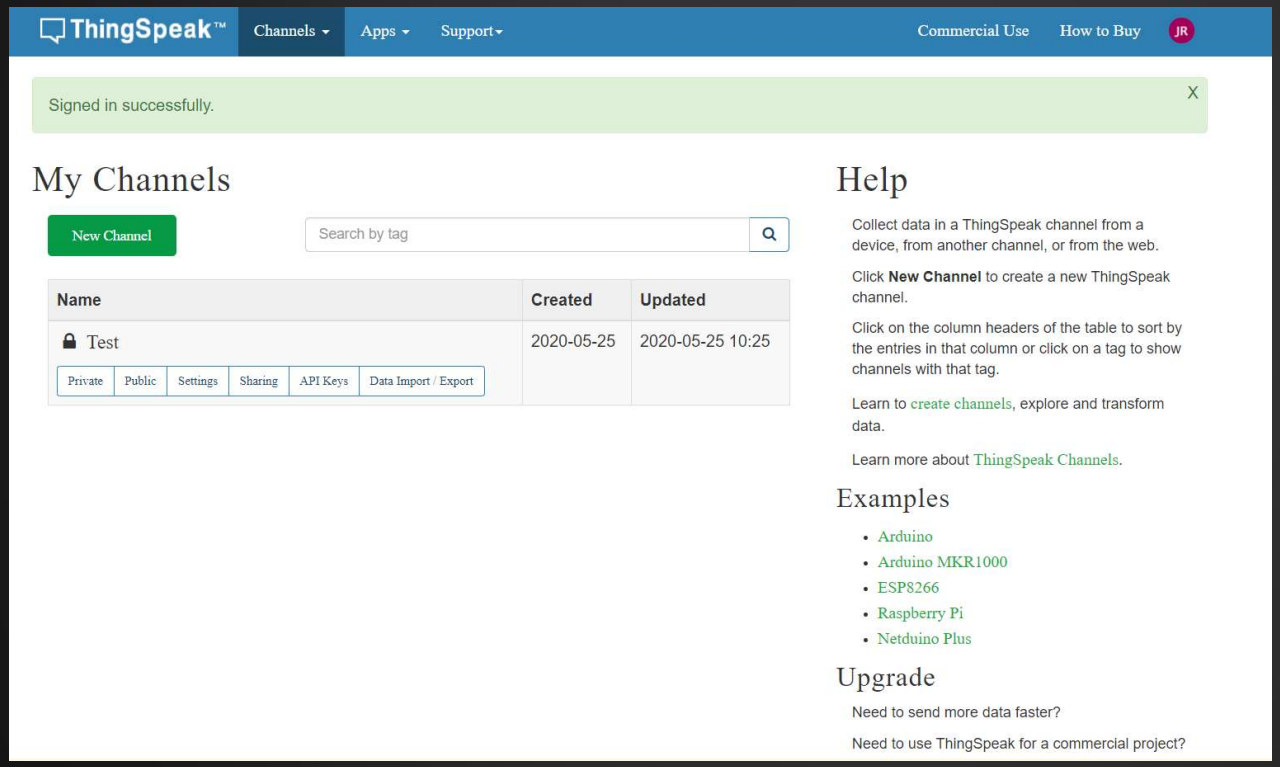
CONNECTION DETAILS

//PB6->TX->ESP8266 RX
//PB7->RX ->ESP8266 TX
//RELAY -PA5

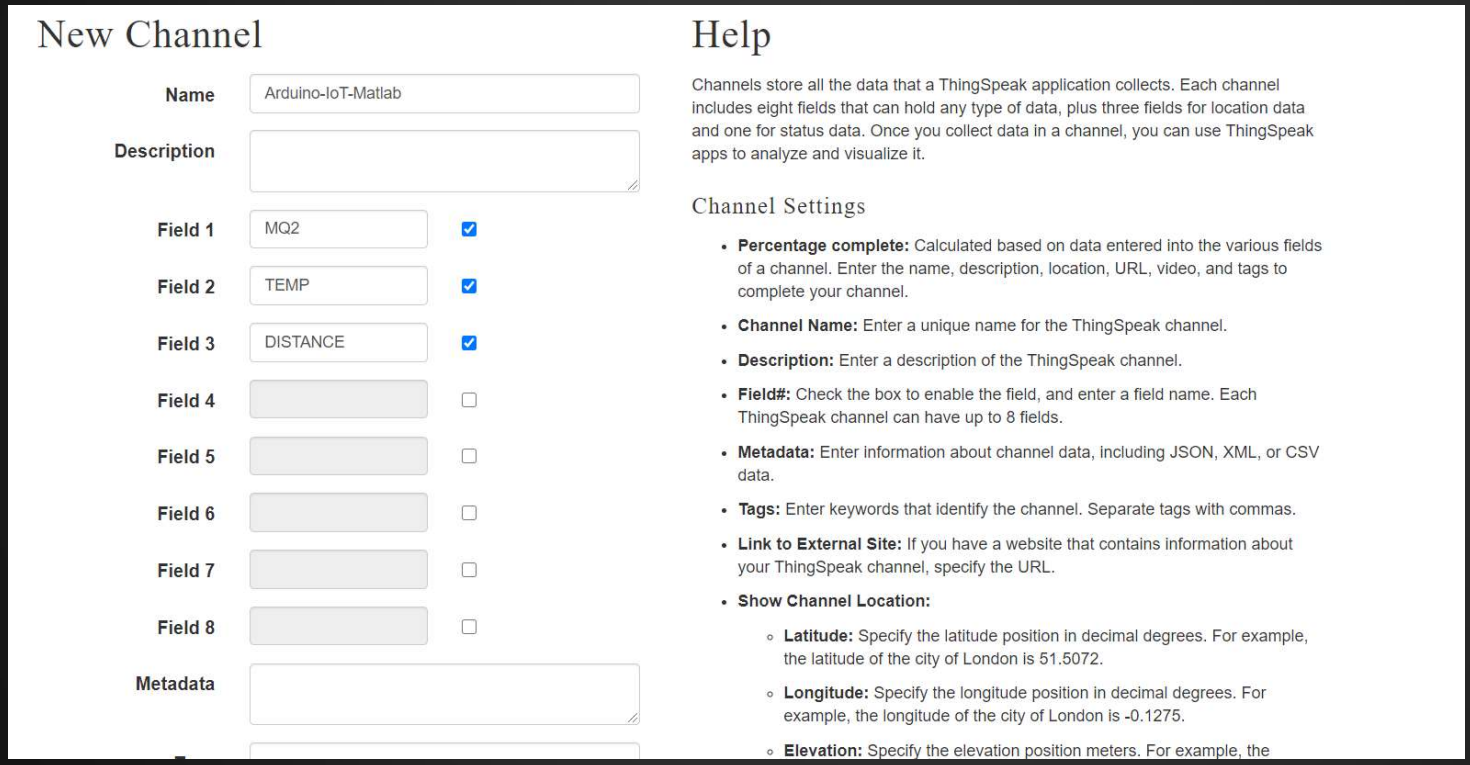
HOME PAGE


ThingSpeak™
Channels
Apps
Support
Commercial Use
How to Buy
JR
Signed in successfully.
X
My Channels
New Channel
Search by tag
Name
Created
Updated
Test
2020-05-25
2020-05-25 10:25
Private
Public
Settings
Sharing
API Keys
Data Import / Export
Help
Collect data in a ThingSpeak channel from a device, from another channel, or from the web.
Click New Channel to create a new ThingSpeak channel.
Click on the column headers of the table to sort by the entries in that column or click on a tag to show channels with that tag.
Learn to create channels, explore and transform data.
Learn more about ThingSpeak Channels.
Examples
Arduino
Arduino MKR1000
ESP8266
Raspberry Pi
Netduino Plus
Upgrade
Need to send more data faster?
Need to use ThingSpeak for a commercial project?

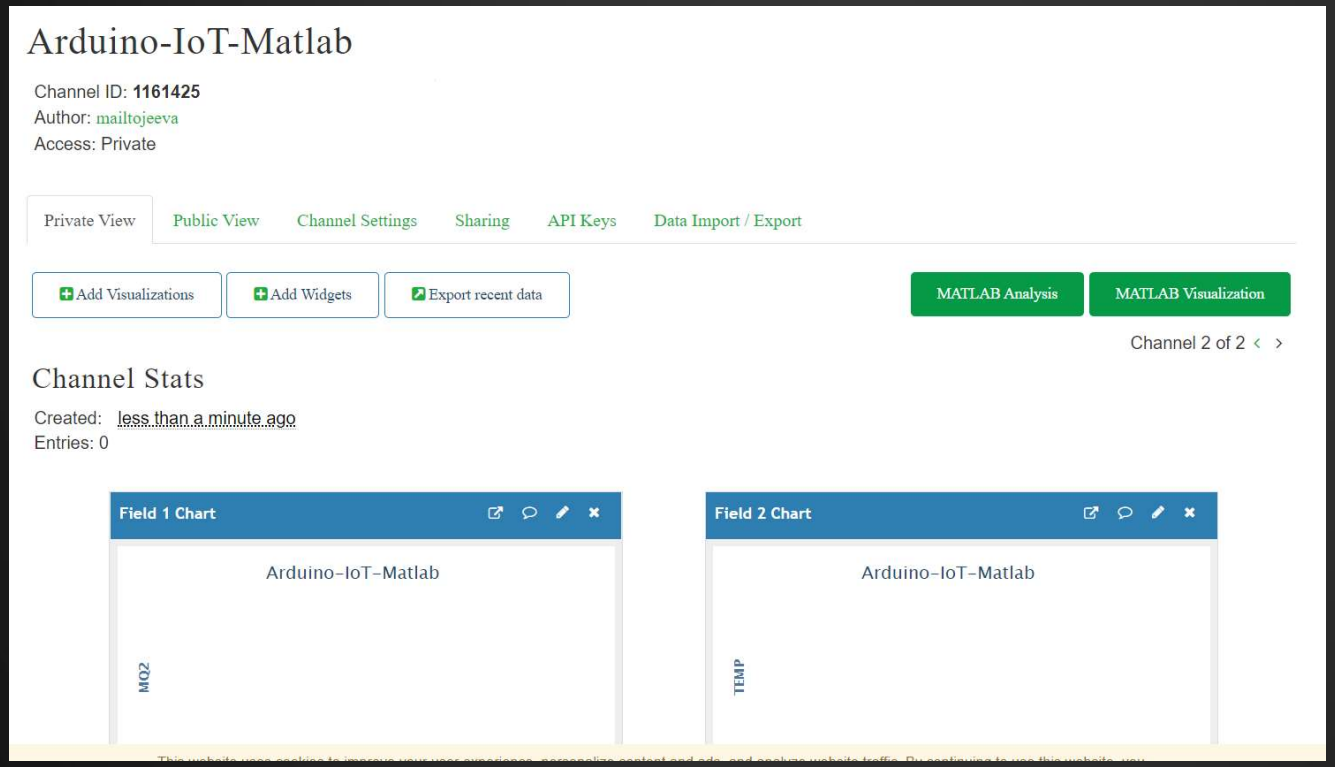
CHANNEL CREATION



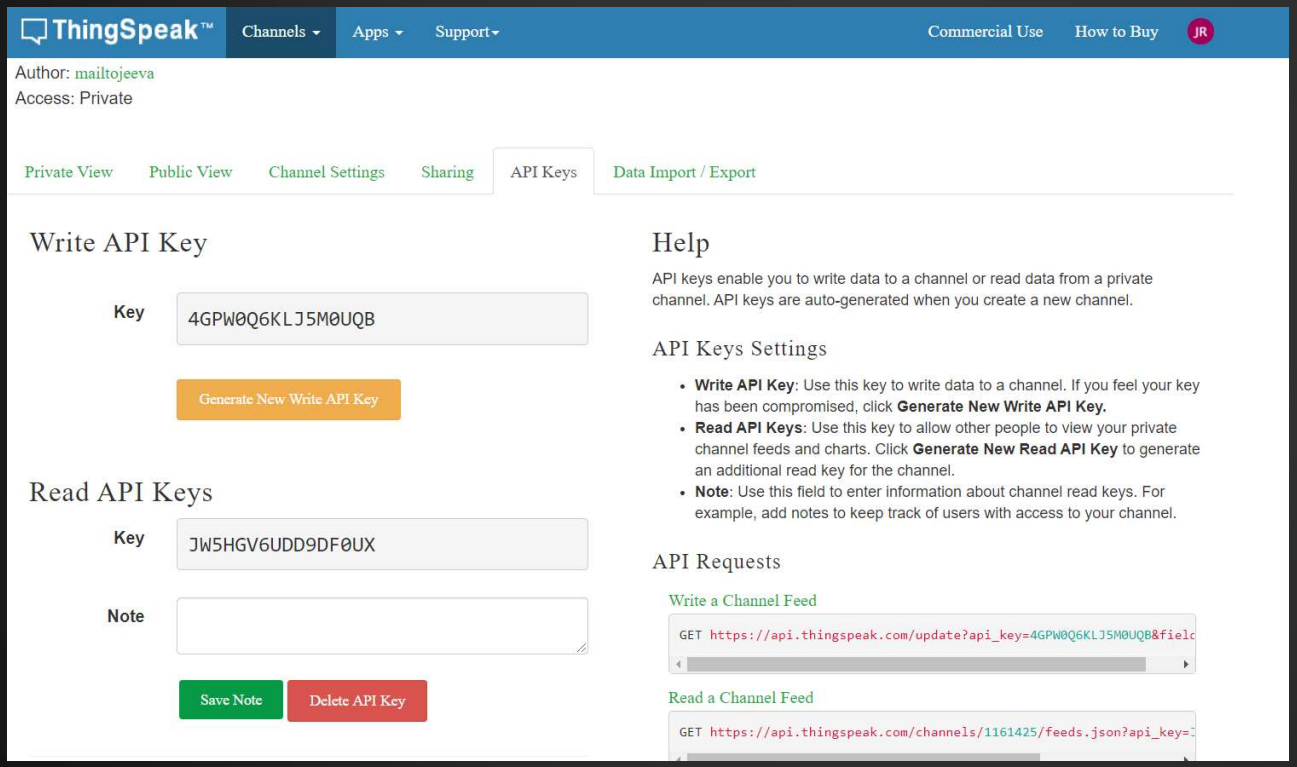
CREATE FIELDS



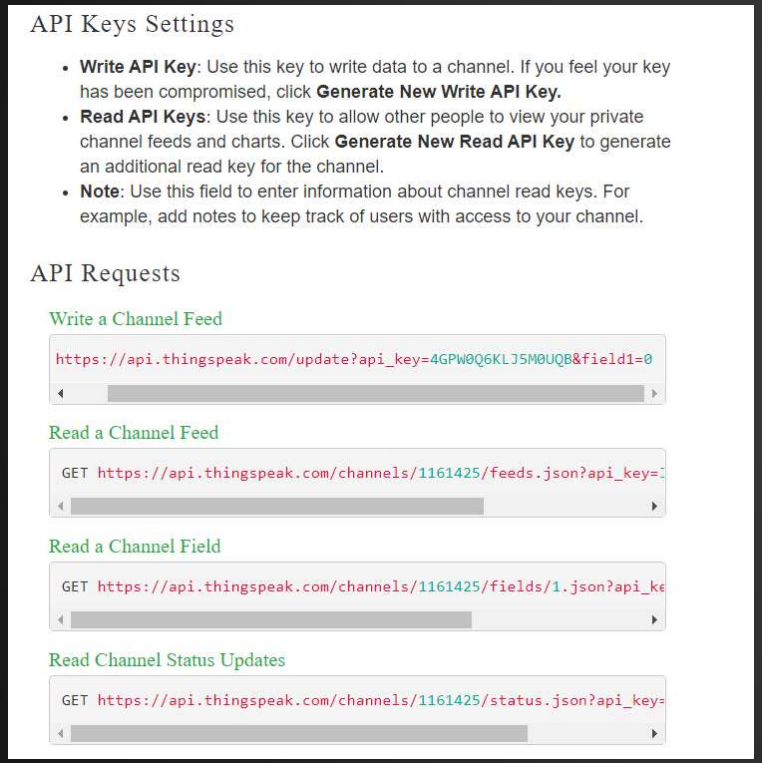
CHANNEL CREATED



READ AND WRITE API KEYS



API KEY SETTINGS



API Keys Settings

- **Write API Key**: Use this key to write data to a channel. If you feel your key has been compromised, click **Generate New Write API Key.**
- **Read API Keys**: Use this key to allow other people to view your private channel feeds and charts. Click **Generate New Read API Key** to generate an additional read key for the channel.
- **Note**: Use this field to enter information about channel read keys. For example, add notes to keep track of users with access to your channel.

API Requests

Write a Channel Feed

```
https://api.thingspeak.com/update?api_key=4GPW0Q6KLJ5M0UQB&field1=0
```

Read a Channel Feed

```
GET https://api.thingspeak.com/channels/1161425/feeds.json?api_key=
```

Read a Channel Field

```
GET https://api.thingspeak.com/channels/1161425/fields/1.json?api_ke
```

Read Channel Status Updates

```
GET https://api.thingspeak.com/channels/1161425/status.json?api_key=
```

DEMO

Mindset Activity

- ✓ Write Down Your Top 10 Goals.(1 Mark)
- ✓ Write Down Your Top 10 Ideas to Achieve Your Goal.(1 Mark)
- ✓ 30 Minutes for Workout (5000-7000 Steps a Day)(2 Mark).
- ✓ 15 Minutes to Meditate (2 Mark)
- ✓ 10 Minutes to Visualize of Achieving Your Goals(1)
- ✓ 10 Minutes to Focus on the Day Plan (1)
- ✓ 2 Hr's for Learning and Take Notes. (2 Mark)

10/10

https://www.facebook.com/groups/embeddedsystemsandiot/

THANK YOU