Midterm 2

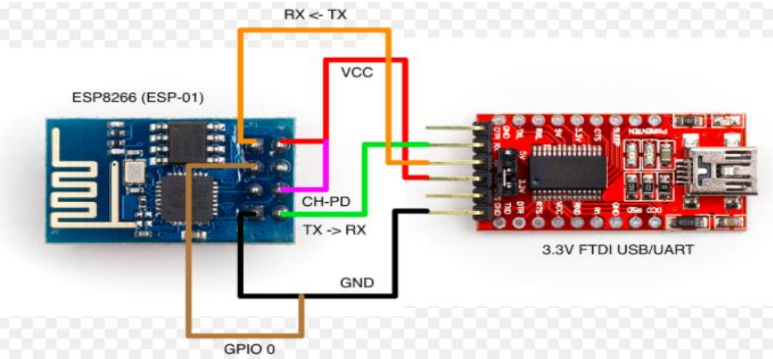
V Muthukumar

The Goal

- Write, simulate, and demonstrate using ATMEL Studio 7 an C code for the AVR ATMEGA328p microcontroller that performs the following functions:
    1. Program the I2C of ATmega328/p to read the I2C sensor (APDS 9960). (based of DA6). Read only the Ambient Light value & RGB Values.
    2. Display the value to UART. (DA3)
    3. Make sure the AT Firmware is downloaded into the ESP-01 module.
    4. Register for a free Thingspeak account with MATHWORK. Setup and get the channel Key.
    5. Transmit sensor value to ESP-01 through UART port using AT Commands.
    6. Display the sensor values as separate graphs in Thingspeak

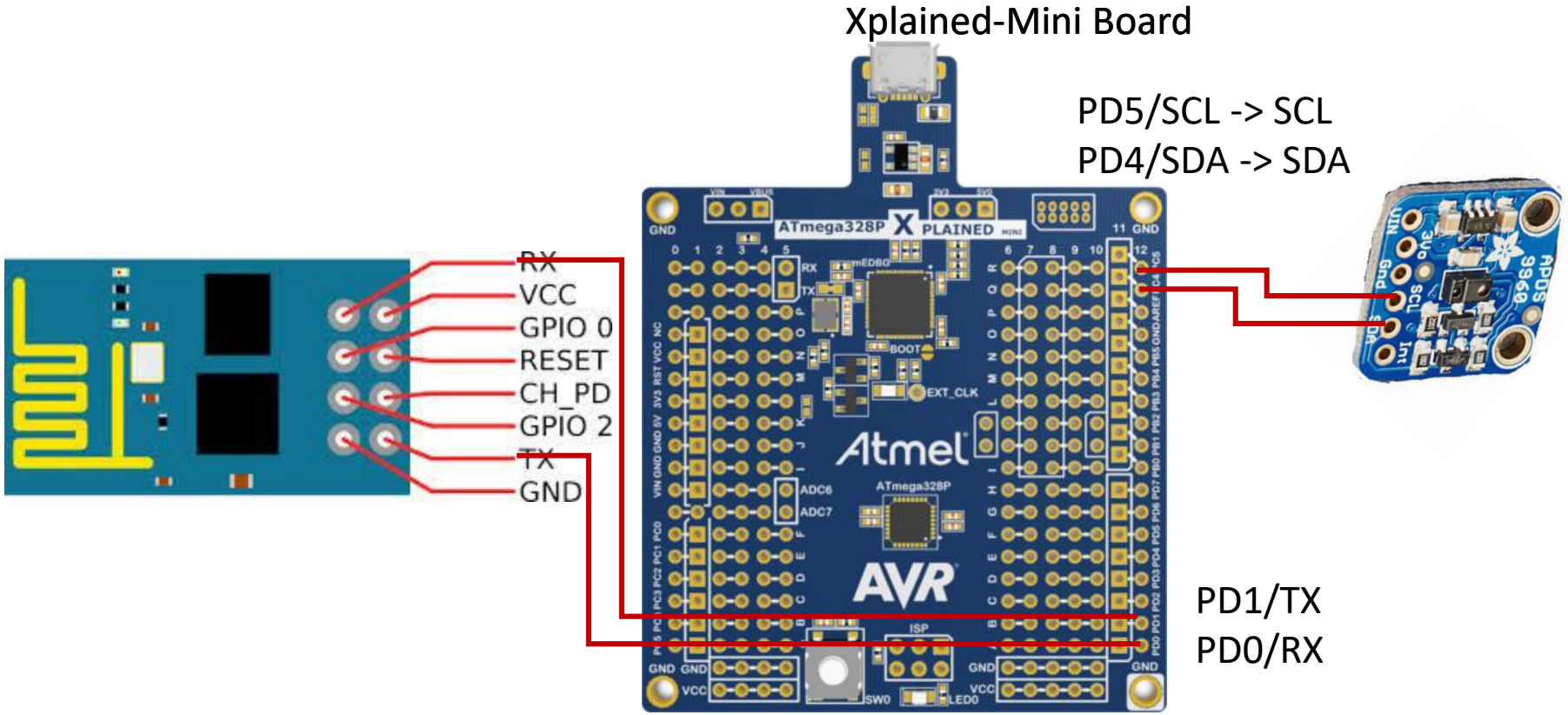
Working with ESP8266-01 Module

- Make sure the AT Firmware is downloaded into the ESP-01 module.
- Follow documentation @ class website
- Firmware:
  - Download Link for Latest Ai-Thinker firmware: http://www.electrodragon.com/w/ESP8266_AT_Commands
  - Alternative Download Link for older version Ai-Thinker firmware: http://wiki.aprbrother.com/wiki/Firmware_For_ESP8266
- Use Flash Software
  - ESP Flash Download Tool (Software) - http://bbs.espressif.com/viewtopic.php?f=57&t=433

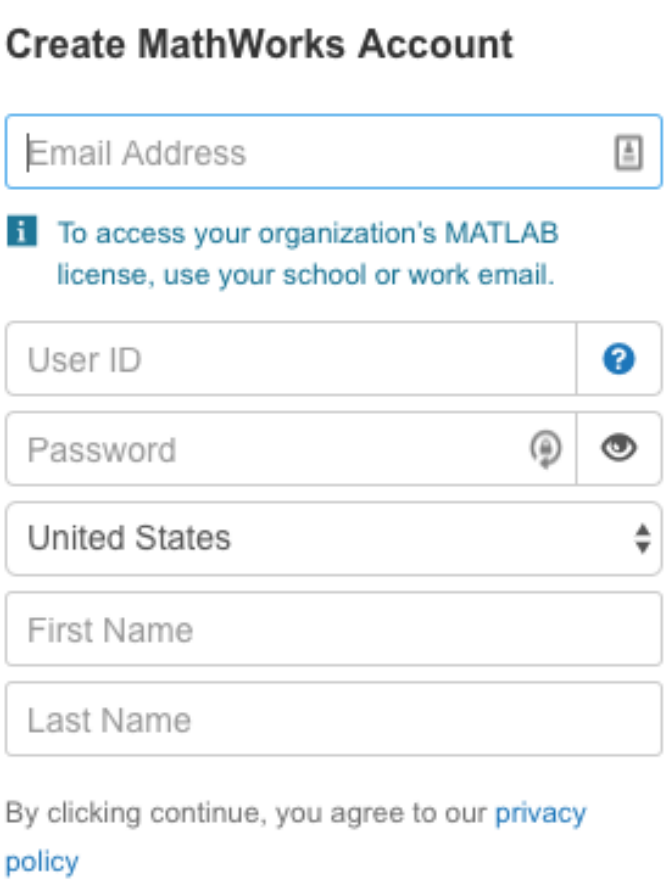


Use only 3.3V from the adapter!!!

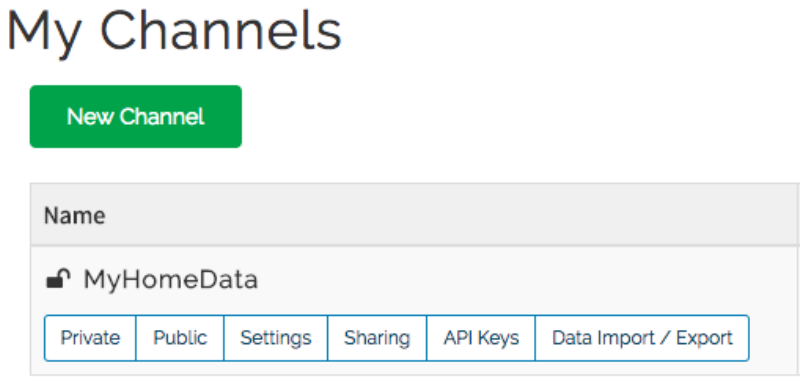
Connecting to ATMega328p



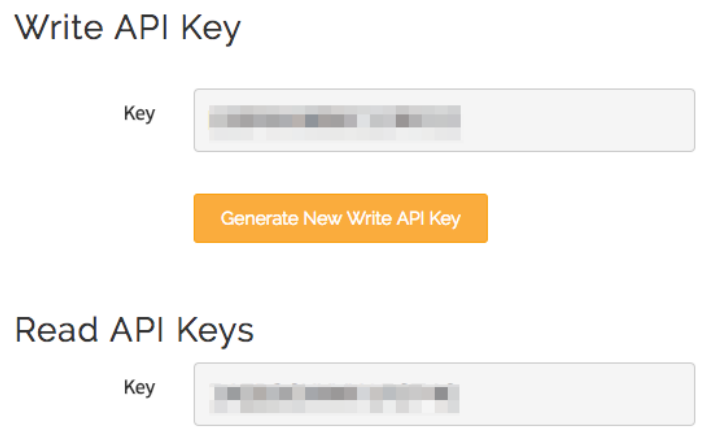
Sign up for ThingSpeak



Add new channel

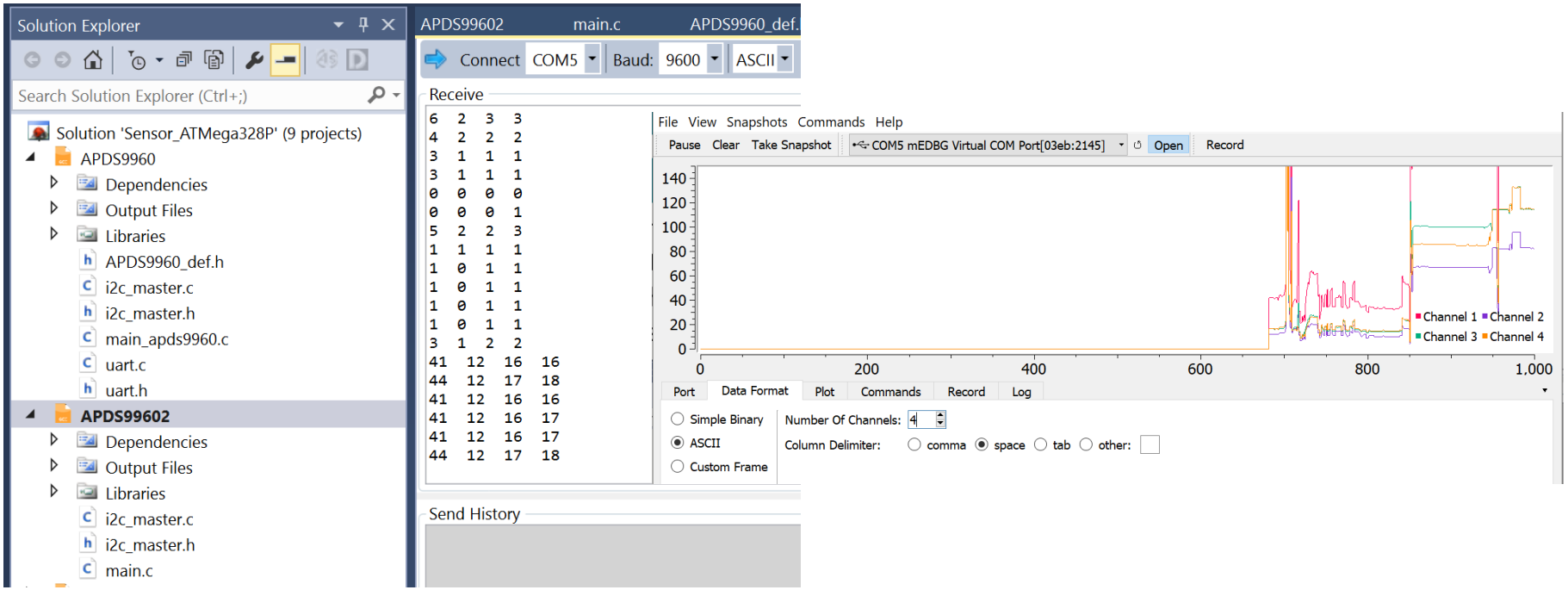


Get API keys



Upload Data

- Connect to your WiFi router & transmit sensor value to ESP-01 through UART port using AT Commands.
- Display the sensor values (each Ambient, Red, Green, and Blue) as separate graphs in Thingspeak



PS: Do not upload assignment with personal WiFi SSID & password

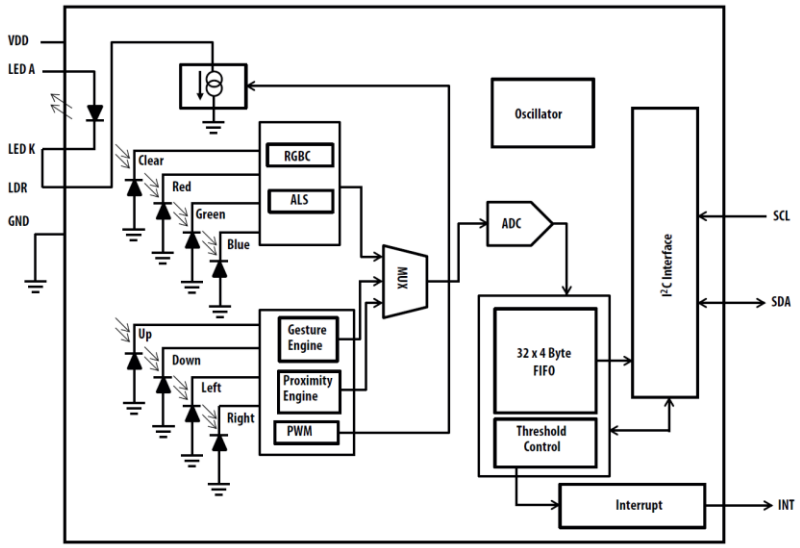
APDS-9960

- The APDS-9960 device features advanced Gesture detection, Proximity detection, Digital Ambient Light Sense (ALS) and Color Sense (RGBC).

- Ambient Light and RGB Color Sensing
  - UV and IR blocking filters
  - Programmable gain and integration time
  - Very high sensitivity – Ideally suited for operation behind dark glass
- Proximity Sensing
  - Trimmed to provide consistent reading
  - Ambient light rejection
  - Offset compensation
  - Programmable driver for IR LED current
  - Saturation indicator bit
- Complex Gesture Sensing
  - Four separate diodes sensitive to different directions
  - Ambient light rejection
  - Offset compensation
  - Programmable driver for IR LED current
  - 32 dataset storage FIFO
  - Interrupt driven I2C-bus communication
- I2C-bus Fast Mode Compatible Interface
  - Data Rates up to 400 kHz
  - Dedicated Interrupt Pin

APDS-9960

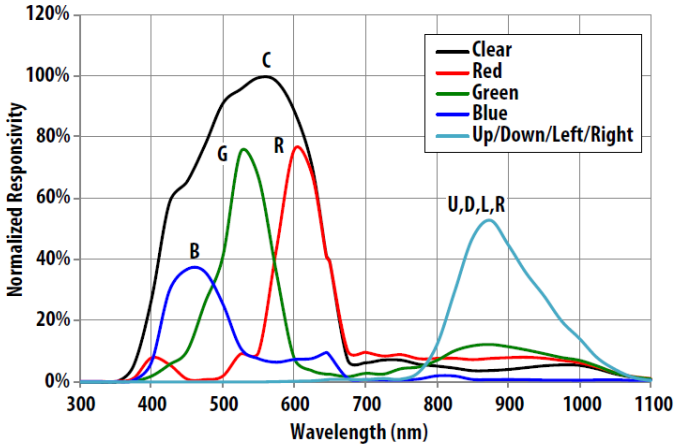
Functional Block Diagram



I/O Pins Configuration

Pin	Name	Type	Description
1	SDA	I/O	I²C serial data I/O terminal - serial data I/O for I²C-bus
2	INT	O	Interrupt - open drain (active low)
3	LDR		LED driver input for proximity IR LED, constant current source LED driver
4	LEDK		LED Cathode, connect to LDR pin when using internal LED driver circuit
5	LEDA		LED Anode, connect to V_{LEDA} on PCB
6	GND		Power supply ground. All voltages are referenced to GND
7	SCL	I	I²C serial clock input terminal - clock signal for I²C serial data
8	V_{DD}		Power supply voltage

Spectral Response



APDS-9960 Register Map

Table 3. Color / ALS Controls 0x80

Register/Bit	Address	Description
ENABLE<PON>	0x80<0>	Power ON
ENABLE<AEN>	0x80<1>	ALS Enable
ENABLE<AIEN>	0x80<4>	ALS Interrupt Enable
ENABLE<WEN>	0x80<3>	Wait Enable
ATIME	0x81	ALS ADC Integration Time
WTIME	0x83	Wait Time
AILTL	0x84	ALS low threshold, lower byte
AILTH	0x85	ALS low threshold, upper byte
AIHTL	0x86	ALS high threshold, lower byte
AIHTH	0x87	ALS high threshold, upper byte
PERS<APERS>	0x8C<3:0>	ALS Interrupt Persistence
CONFIG1<WLONG>	0x8D<1>	Wait Long Enable
CONTROL<AGAIN>	0x8F<1:0>	ALS Gain Control
CONFIG2<CPSIEN>	0x90<6>	Clear diode Saturation Interrupt Enable
STATUS<CPSAT>	0x93<7>	Clear Diode Saturation
STATUS<AINT>	0x93<4>	ALS Interrupt
STATUS<AVALID>	0x93<0>	ALS Valid
CDATAL	0x94	Clear Data, Low byte
CDATAH	0x95	Clear Data, High byte
RDATAL	0x96	Red Data, Low byte
RDATAH	0x97	Red Data, High byte
GDATAL	0x98	Green Data, Low byte
GDATAH	0x99	Green Data, High byte
BDATAL	0x9A	Blue Data, Low byte
BDATAH	0x9B	Blue Data, High byte
CICLEAR	0xE5	Clear Channel Interrupt Clear
AICLEAR	0xE7	All Non-Gesture Interrupt Clear

0x81

Field	Bits	Description			
ATIME	7:0	**FIELD VALUE**	**CYCLES**	**TIME**	**MAX COUNT**
		0	256	712 ms	65535
		182	72	200 ms	65535
		= 256 – TIME / 2.78 ms	…	…	…
		219	37	103 ms	37889
		246	10	27.8 ms	10241
		255	1	2.78 ms	1025

0x8F

AGAIN	1:0	ALS and Color Gain Control.

FIELD VALUE	GAIN VALUE
0	1x
1	4x
2	16x
3	64x

Data: 0x94 – 0x9B

Note:

- Make sure your values don’t saturate
- Extra Points:
  - Work with interrupts
  - Enable threshold
  - Include Proximity function

Code Template

APDS9960
https://github.com/venki666/cpe301Demo/tree/master/Sensors/APDS99602

MPU6050:
https://github.com/venki666/cpe301Demo/tree/master/Sensors/MPU6050_Plot