Research Design Lab

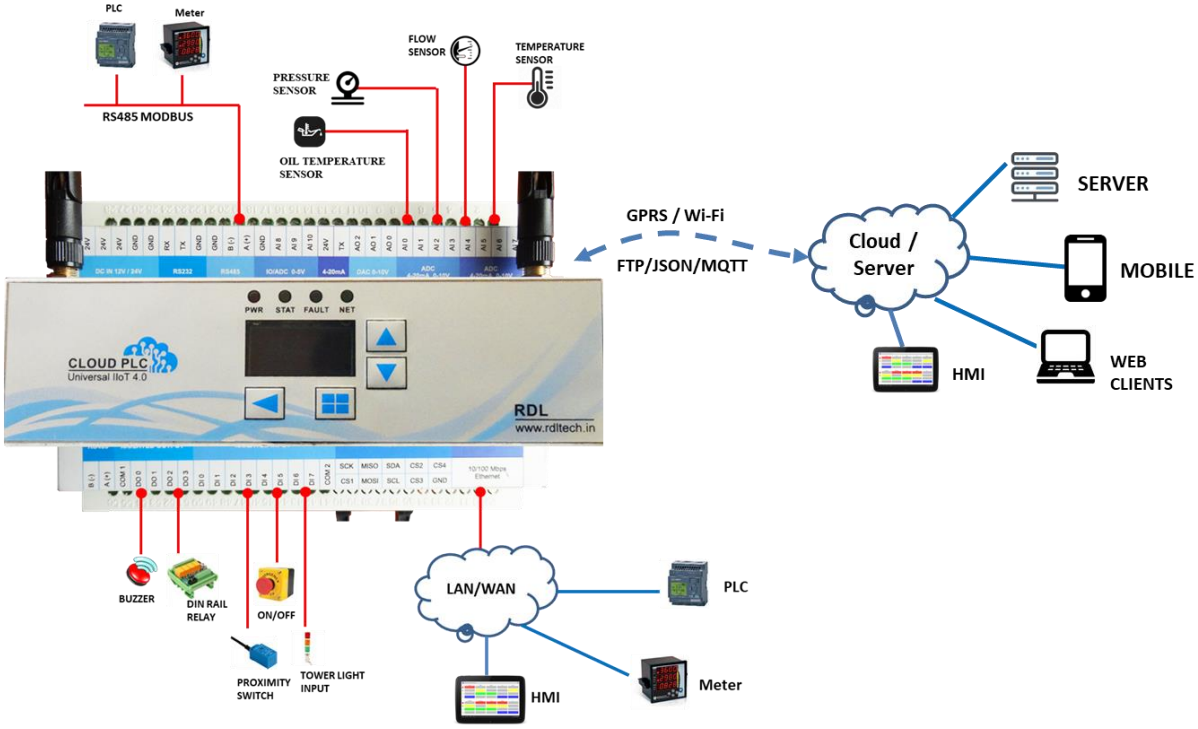
Cloud PLC 4.0

RDL Technologies Pvt Ltd
5th Floor,Sahyadri Campus
Mangaluru-575007
P: +91 824 2988407
sales@researchdesignlab.com
support@researchdesignlab.com

Contents

1. Introduction

Cloud PLC 4.0 series defines a new way of transforming factories into smart/intelligent factories for efficient and easy remote monitoring and controlling the operational status of facilities such as on/off status, pressure and temperature. Cloud PLC 4.0 supporting for custom programming and wide range of industrial protocols like Modbus, MQTT, JSON, RESTful, TCP/UDP, SNMP protocol, which makes the monitoring and solution integration, easier than ever for IT engineers through open source APIs.



Cloud PLC 4.0 can used to build the custom industrial solution for monitoring and controlling PLC and SCADA, HMI, VFD, Motors, servo, Valves, energy meter, actuators, relays, encoder, RFID and fingerprint readers, and industrial sensors.

2. Features

- **Digital IO**
- 24v 3x Isolated Digital output
- 24v 3x Isolated Digital Input
- Isolation: 3750VRMS
- **Analog IO**
- 8x ADC 0-10V/ 4-20mA max.
- 10 / 16 bit ADC resolution
- 3xDAC 0-10V
- 1x 4-20ma TX
- 3xADC 0-5v
- **Wired Connectivity**
- RS485 MODBUS, RS232 & USB
- Ethernet 10/100Mbps, RDL Expansion Bus
- **Memory**
- FRAM 25KB, SD CARD 16GB
- **Wireless connectivity**
- Wi-Fi: 802.11 b/g/n/e/i (802.11n @ up to 150 Mbit/s) Bluetooth: v4.2 Bluetooth Low Energy (BLE) LoRa / Xbee, GPRS/GPS
- **RTC**
- Built-in RTC for stamped data logging
- **Protocol**
- TCP-IP, UDP, MODBUS, FTP, RESTFULL, JSON & MQTT
- **Security:** SSL
- **Power supply**
- DC 12 - 24v
- **Enclouser**
- IP 20
- mounting: Wall / DIN Rail
- Dimension (LxWxH): 155x82x58.5

3. Benefits

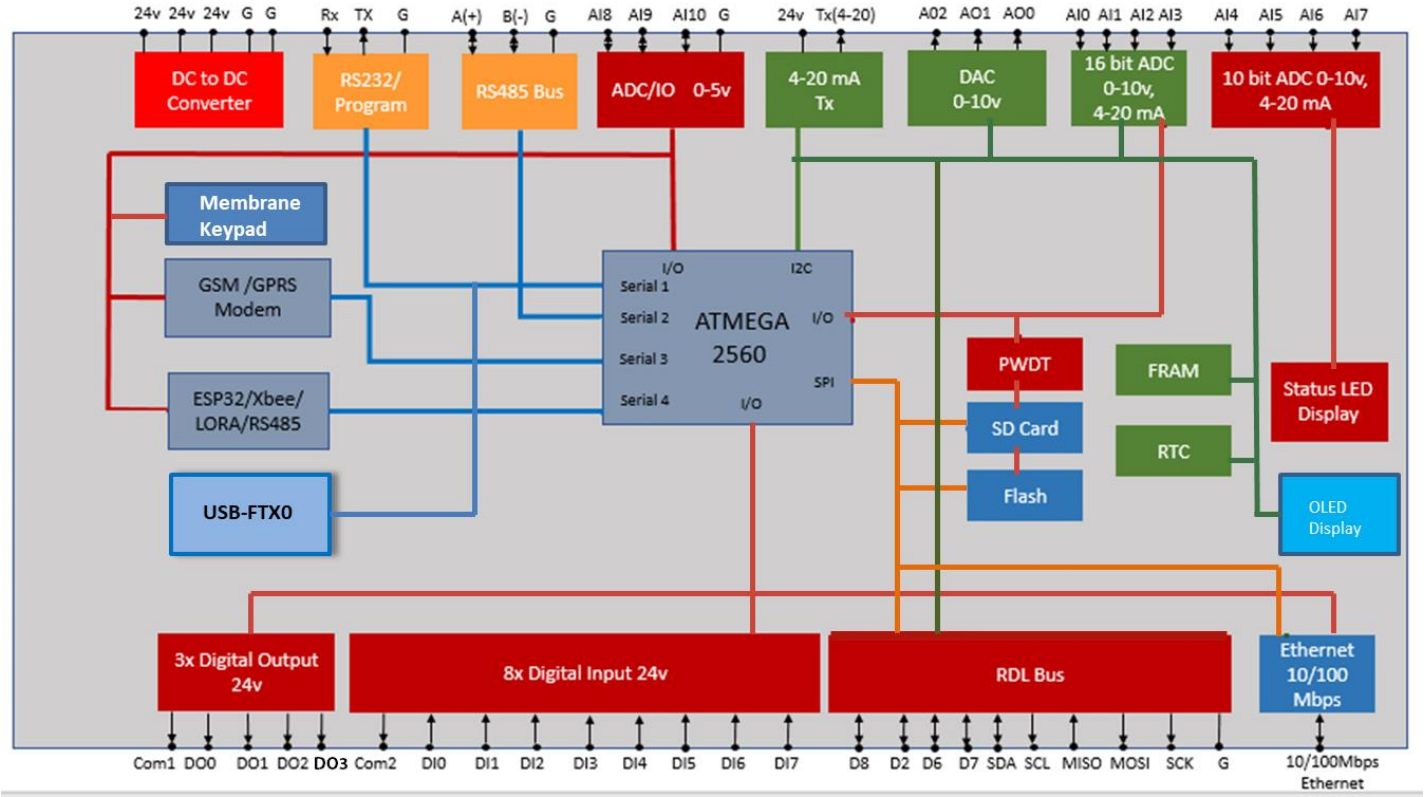
Cloud PLC 4.0 can used to build the custom industrial solution for monitoring and controlling PLC and SCADA, HMI, VFD, Motors, Servo, Valves, Energy meter, Actuators, relays, encoder , RFID and fingerprint readers, industrial sensors and many more with below mentioned operational benefits.

- Simplified logging network as RDL Data Logger supports multiple features
- Paper-less Production environment
- Production count, rejections
- Machine availability and Downtimes
- Preventive maintenance
- Performance Forecasting
- Enable Management by IT

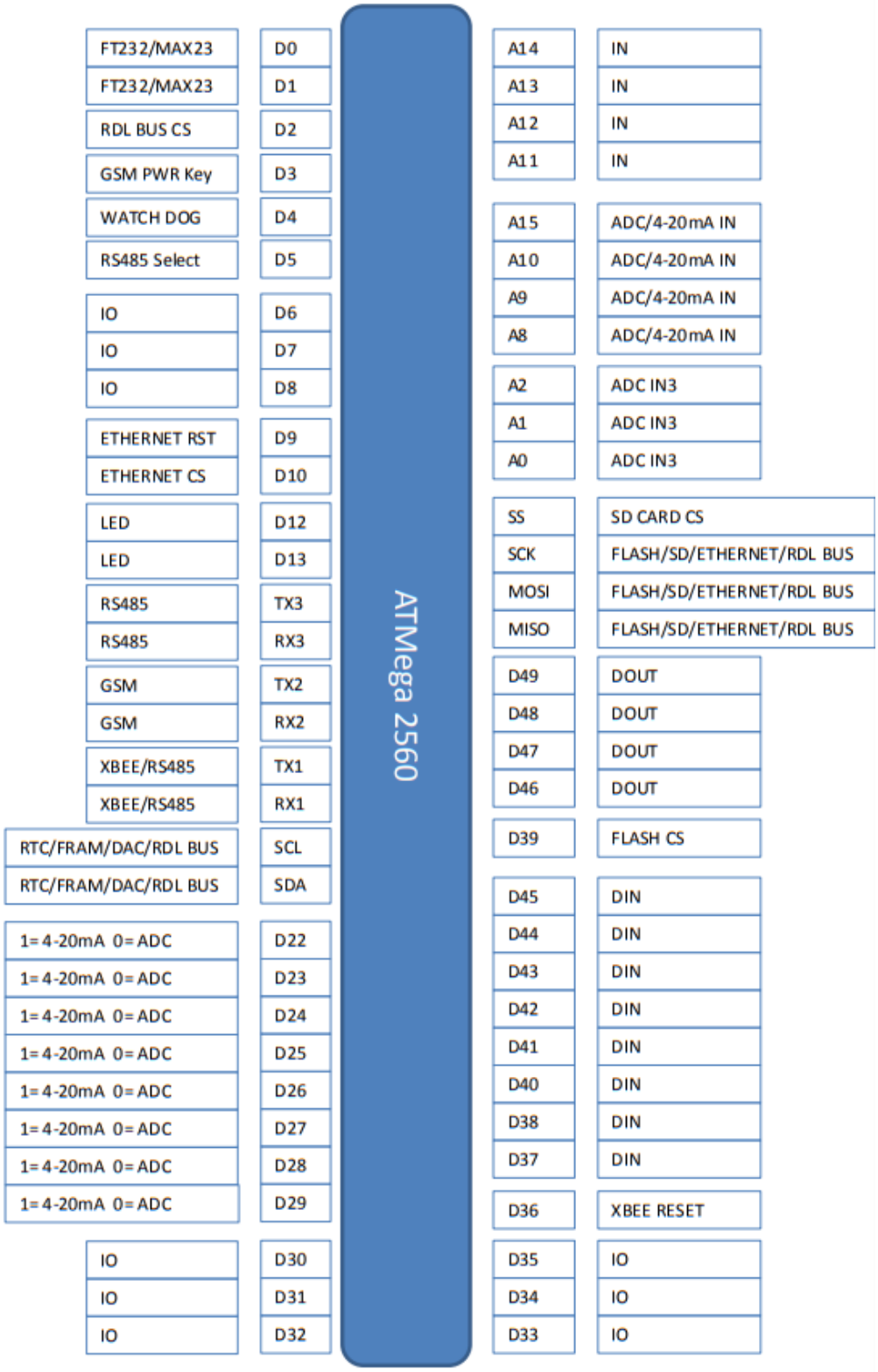
4. Applications

- Production and process monitoring.
- Utilities monitoring.
- Condition monitoring.
- Environment monitoring.
- Industrial Smart grid
- Leakage detection.
- Cold storage monitoring.
- District metering.
- Water treatment.
- Generator monitoring.
- Green House.
- Warning message in case of calamities.
- Standard SCADA Applications

5. Block Diagram



6. Pin-out


FT232/MAX23 D0
FT232/MAX23 D1
RDL BUS CS D2
GSM PWR Key D3
WATCH DOG D4
RS485 Select D5
IO D6
IO D7
IO D8
ETHERNET RST D9
ETHERNET CS D10
LED D12
LED D13
RS485 TX3
RS485 RX3
GSM TX2
GSM RX2
XBEE/RS485 TX1
XBEE/RS485 RX1
RTC/FRAM/DAC/RDL BUS SCL
RTC/FRAM/DAC/RDL BUS SDA
1=4-20mA 0=ADC D22
1=4-20mA 0=ADC D23
1=4-20mA 0=ADC D24
1=4-20mA 0=ADC D25
1=4-20mA 0=ADC D26
1=4-20mA 0=ADC D27
1=4-20mA 0=ADC D28
1=4-20mA 0=ADC D29
IO D30
IO D31
IO D32
ATMega 2560
A14 IN
A13 IN
A12 IN
A11 IN
A15 ADC/4-20mA IN
A10 ADC/4-20mA IN
A9 ADC/4-20mA IN
A8 ADC/4-20mA IN
A2 ADC IN3
A1 ADC IN3
A0 ADC IN3
SS SD CARD CS
SCK FLASH/SD/ETHERNET/RDL BUS
MOSI FLASH/SD/ETHERNET/RDL BUS
MISO FLASH/SD/ETHERNET/RDL BUS
D49 DOUT
D48 DOUT
D47 DOUT
D46 DOUT
D39 FLASH CS
D45 DIN
D44 DIN
D43 DIN
D42 DIN
D41 DIN
D40 DIN
D38 DIN
D37 DIN
D36 XBEE RESET
D35 IO
D34 IO
D33 IO

6.1 Pin Mapping:

Function Blocks	Screw Terminal (External)	On board Connection (Internal)	Reference Pin MEGA 2560	Reference Pin Atmel Studio
MAX232	TX		D0	PE0
	RX		D1	PE1
RS485	A (D+)		TX3	PJ1
	B (D-)		RX3	PJ0
		Enable	D5	PE3
Physical Watch Dog Timer max 3 minutes		Pulse Input	D4	PG5
Opto Isolated Input 24V	DI0		D45	PL4
	DI1		D44	PL5
	DI2		D43	PL6
	DI3		D42	PL7
	DI4		D38	PD7
	DI5		D41	PG0
	DI6		D40	PG1

	DI6		D40	PG1
	DI7		D37	PC0
Opto Isolated Output PWM/Output 24V	DO1		D49	PL0
	DO2		D48	PL1
	DO3		D47	PL2
	DO4		D46	PL3
SD Card		MISO	MISO	PB3
		MOSI	MOSI	PB2
		SCK	SCK	PB1
		SD_CS	SS	PB0
FLASH 8MB		MISO	MISO	PB3
		MOSI	MOSI	PB2
		SCK	SCK	PB1
		SRAM_CS	D39	PG2
FT230x-USB		TX0	D0	PE0
		RX0	D1	PE1
FRAM		SCL	SCL	PD0
		SDA	SDA	PD1
RTC		SCL	SCL	PD0
		SDA	SDA	PD1
ESP32/Xbee /LoRa/BLE		TX3	TX1	PD3
		RX3	RX1	PD2
		RESET	D36	PC1

GSM/GPRS		TX2	TX2	PH1
		RX2	RX2	PH0
		GSM POWER KEY	D3	PE5
RDL BUS	SCL		SCL	PD0
10 Pin FRC Connector	SDA		SDA	PD1
	MISO		MISO	PB3
	MOSI		MOSI	PB2
	SCK		SCK	PB1
	CS1		D2	PE4
	CS2		D6	PH3
	CS3		D7	PH4
	CS4		D8	PH5
ETHERNET	RJ45	SCK	SCK	PB1
		MOSI	MOSI	PB2
		MISO	MISO	PB3
		RST	D9	PH6
		CS	D10	PB4
Analog ADC/4-20MA Receiver 16 bit Selection Pin 0 ---- Analog 1 ---- 4-20MA		A1	D26	PA4
		A2	D22	PA0
		A3	D27	PA5
		A4	D23	PA1
Analog 10 Bit Selection Pin 0 ---- Analog 1 ---- 4-20MA		A5	D28	PA6
		A6	D24	PA2
		A7	D25	PA3
		A8	D29	PA7

Analog ADC 0-10v / 4-20MA Receiver 16 Bit I2C	AI0	SCL	D20	PD0
	AI1	SDA	D21	PD1
	AI2			
	AI3			
ADC 0-10V /4-20MA 10 Bit	AI4		A10	PK2
	AI5		A9	PK1
	AI6		A8	PK0
	AI7		A15	PK7
DAC 0-10V	AO0	SCL	D20	PD0
	AO1	SDA	D21	PD1
	AO2			
4-20MA TX	TX	SCL	D20	PD0
		SDA	D21	PD1
Status LED	PWR	5V		
	STAT	GSM STAT		
	FAULT		D13	PB7
	NET		D12	PB6
OLED		SCL	D20	PD0
		SDA	D21	PD1
Membrane Keypad	ENTER		D30	PC7
	UP		D31	PC6
	DOWN		D32	PC5
	MENU		D33	PC4
IO/ADC 0-5V	AI8		ADC0	PF0
	AI9		ADC1	PF1
	AI10		ADC2	PF2

7. Programming IDE

The hardware supports various Open Source Programming IDE including Arduino IDE, Atmel Studio and Arduino Compatible Compiler for Lab View. For more information on this follow "**Open Source Programming IDE"** section of the following link.

https://rdltech.in/cloud-plc-4-0

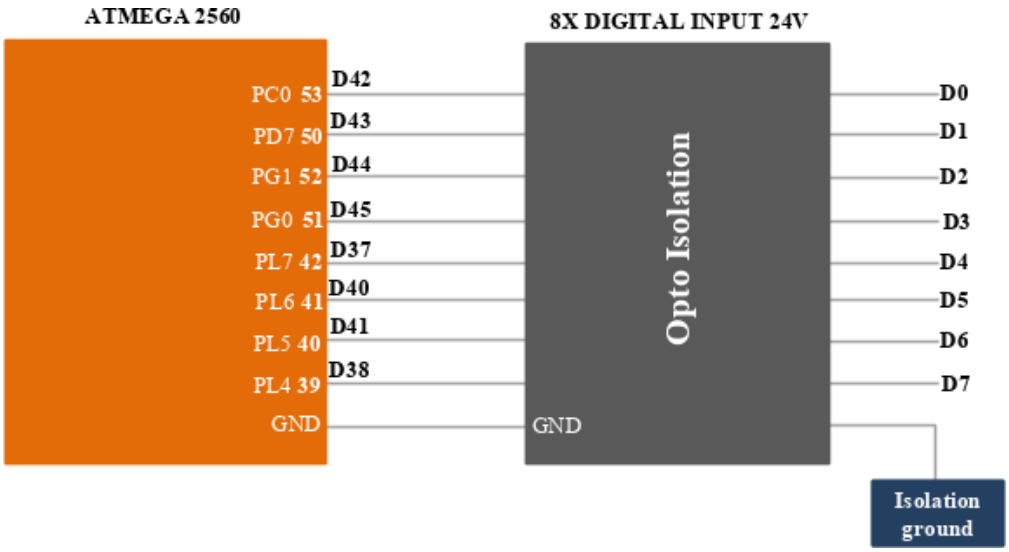
8. Product Specification

8.1. Digital Input

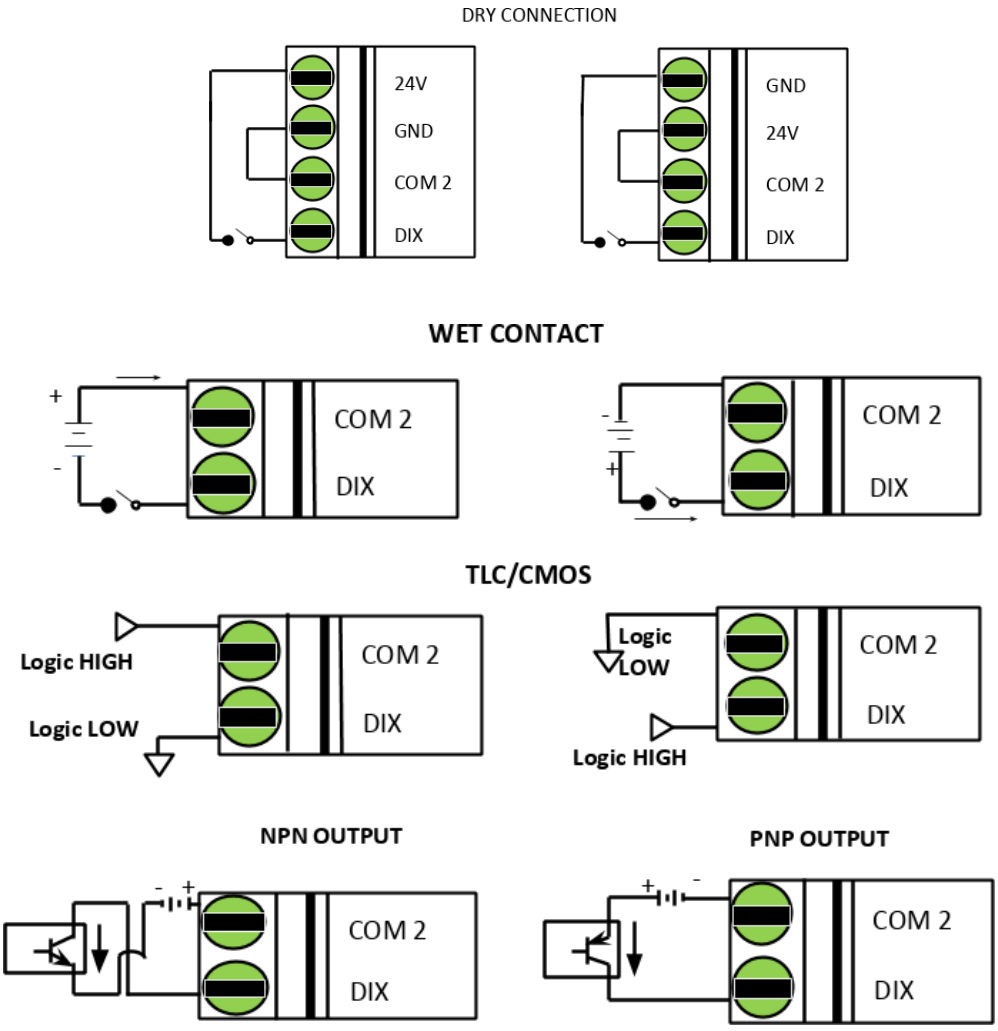
Specification

- Channels: 8
- Input Voltage: 0-24V
  - Logic High: >11V
  - Logic Low: <3V
- Isolation : 3750 VRMS
- Supports Inverted DI Status
- Supported Connection: Dry and Wet both
- Maximum Frequency : 200Hz-38KHz

Functional Diagram

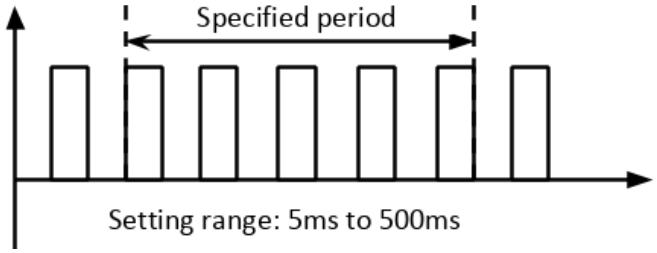


Application Wiring



Use Case

1. Measuring Frequency



Example Code

You may look into the following link for example on reading a digital pin.

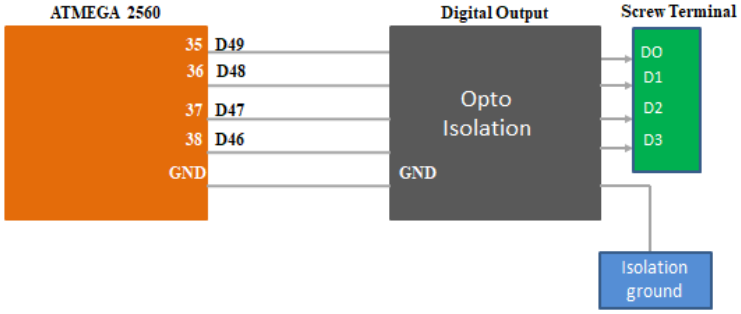
https://www.arduino.cc/reference/en/language/functions/digital-io/digitalread/

8.2. Digital Output

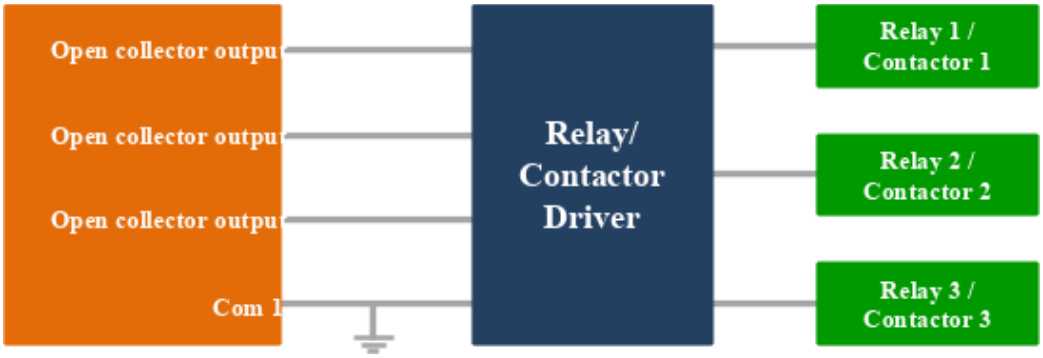
Specification

- Channels: 3
- Open Collector
- Isolation : 3750 VRMS
- Absolute maximum voltage - 35V, Current - 100mA
- Cut-Off Frequency : 10KHz

Functional Diagram



Application Wiring



Example Code

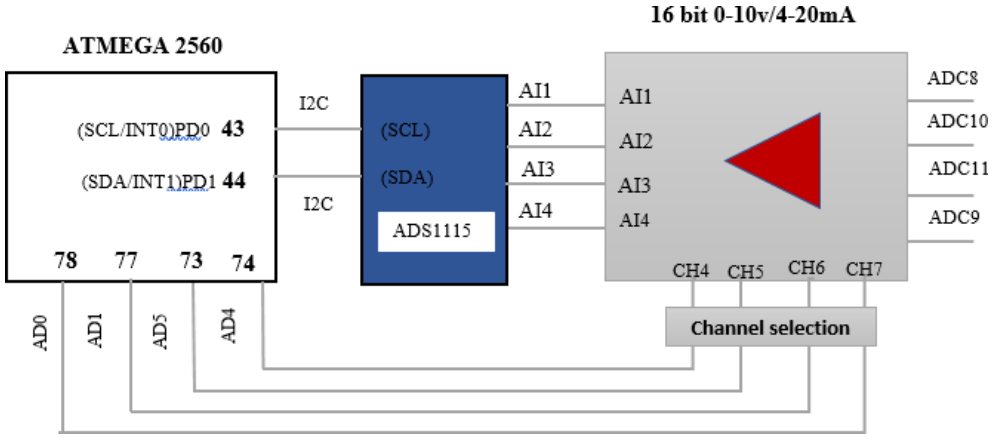
You may look into the following link for more details on writing to digital pin

https://www.arduino.cc/en/Reference.digitalWrite

8.3. Analog Input

Specification

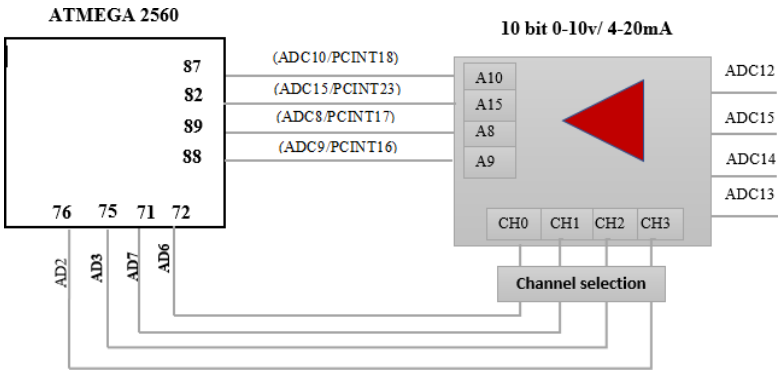
- Channels: 8+3
  Group 1:
  - Channel : 4
  - Input : Voltage(0-10V) / Current(4-20mA)
  - Resolution : 16 bits



- Sampling Rate – 860 sample/sec

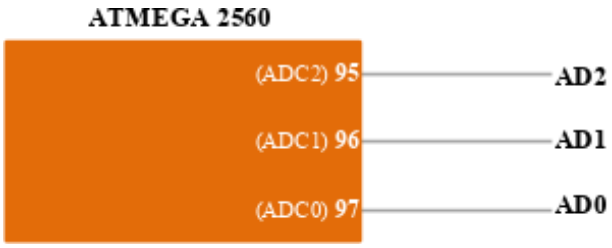
Group 2:

- Channel : 4
- Input : Voltage(0-10V) / Current(4-20mA)
- Resolution : 10 bits
- Sampling Rate – 9.6KHz [13 clocks]



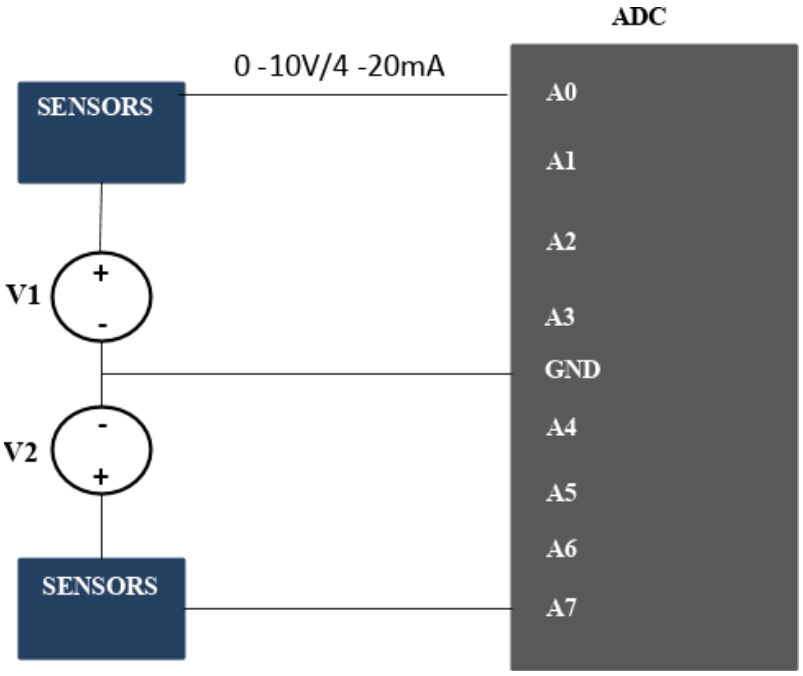
Group 3:

- Channel : 3
- Input : Voltage(0-5V)
- Resolution : 10 bits
- Sampling Rate – 9.6KHz (13 clocks)



Application Wiring

1. Interfacing ADC with Sensor



ATTENTION: When an ADC Channel is configured for measuring loop current, voltage source should never be given to the channel. If given, damage could happen to the internal circuitry or external device connected to it.

Example Code

You may look into the following link for more details on reading analog pin.

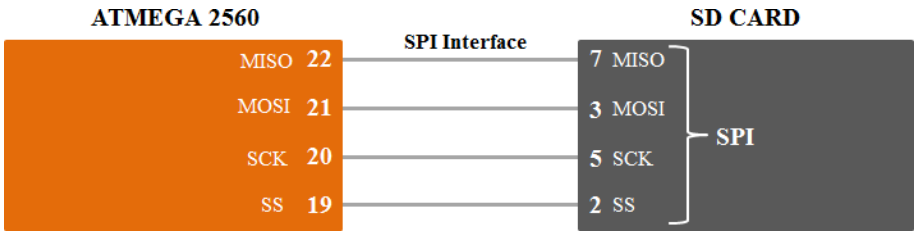
https://www.arduino.cc/en/Tutorial/ReadAnalogVoltage#toc5

8.4. SD Card

Specification

- SPI Serial Interface
- Supports Fat File system

Functional Diagram



Example Code

You may look into the following link for example on SD Card.

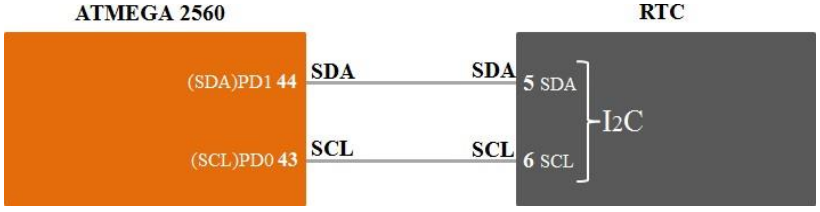
https://www.arduino.cc/en/Reference/SD

8.5. RTC

Specification

- DS1307 with I2C Serial Interface
- Counts Seconds, Minutes, Hours, Date, Month, Day, and Year with Leap-Year Compensation.
- 56-Byte, Battery-Backed, NV RAM for Data Storage
- Consumes <500nA in Battery Backup Mode with Oscillator Running

Functional Diagram



Example Code

You may look into the following link for example code on RTC.

https://www.arduino.cc/en/Reference/RTC

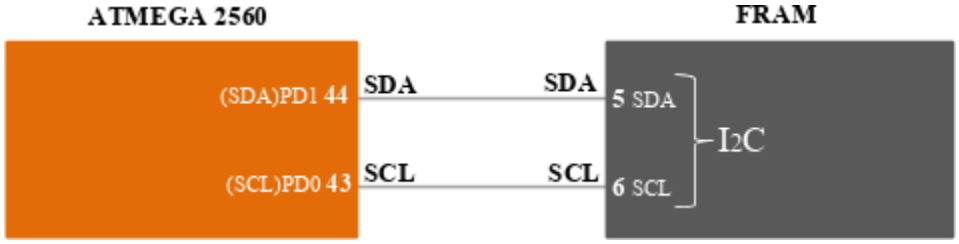
8.6. FRAM

FRAM is specifically used for applications such as production counting, production rejection where variable subjected to continuous write cycle

Specification

- MB85RC256V, I2C compatible with Bit configuration : 32,768 words × 8 bits
- Operating frequency : 1 MHz (Max)
- Read/write endurance : 1012 times / byte
- Number of write cycles: 100 Trillion times
- Operating power supply voltage : 2.7V to 5.5V, current 200 µA
- Data Retention: 10 years (+85°C), 95 years (+55°C), over 200 years (+35°C).

Functional Diagram



Example Code

You may look into the following link for example on RTC

https://github.com/adafruit/Adafruit_FRAM_I2C/blob/master/examples/MB85RC256V/MB85RC256V.ino

8.7. PWDT (Physical Watch Dog Timer)

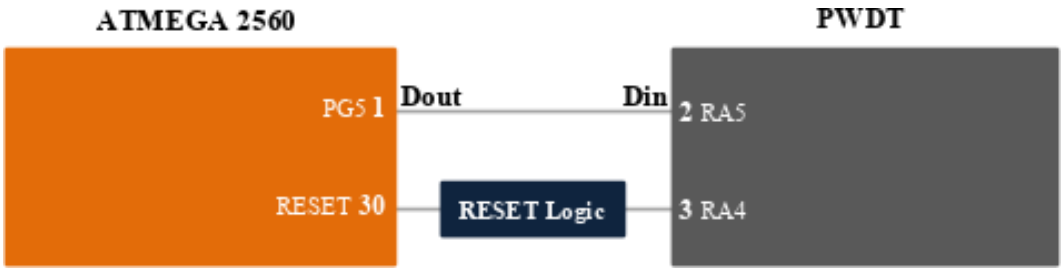
External physical watchdog is connected along with inbuilt watchdog timer. There are many instances where we need to set watch dog time for more than 8 seconds (typically bulk file upload takes in minutes). As inbuilt WDT is limited to maximum of 8Sec, we have gone a step further to support watch dog time up to 3 minutes.

ATTENTION: User must program PWDT to refresh before the timer (3 min) expires.

Specification

- PWDT supports up to 3minutes.
- PIC12F1840 used for PWDT
- Refresh time : 1 pulse in every 3 minutes
- Operating temperature range: –40 to 125 °C

Functional Diagram



Example Code

You may look into the following link for examples on watchdog timer.

https://folk.uio.no/jeanra/Microelectronics/ArduinoWatchdog.html

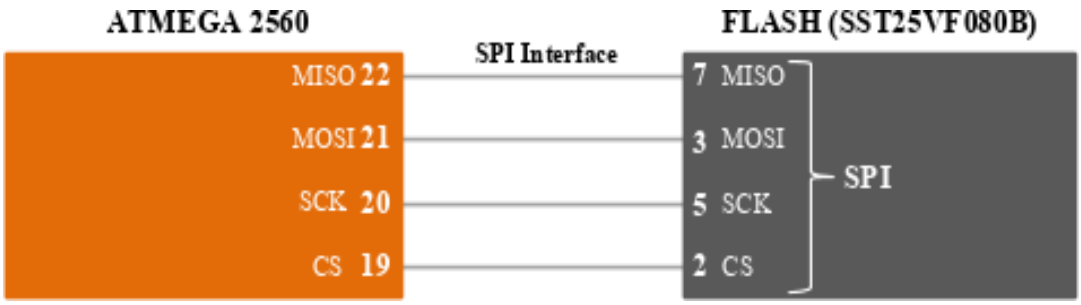
8.8. Flash

Flash is specifically used for embedded server.

Specification

- SST25VF080B - SPI Compatible: Mode 0 and Mode 3
- Memory: 8MBit
- High Speed Clock Frequency – 50 MHz
- Single Voltage Read and Write Operations – 2.7-3.6V
- Endurance: 100,000 Cycles (typical) – >100 years Data Retention
- Low Power Consumption: – Active Read Current: 10 mA (typical)
- Flexible Erase Capability – Uniform 4KB, 32KB overlay blocks and 64KB overlay blocks
- Software Write Protection

Functional Diagram



Example Code

You may look into the following link for example on how to use flash.

https://github.com/nullboundary/SST25VF

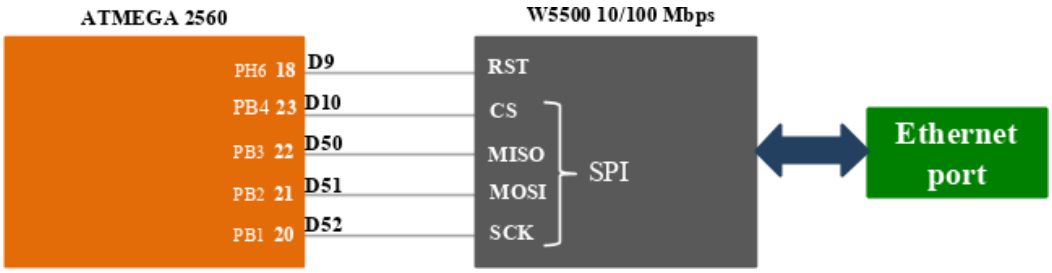
8.9. Ethernet

Ethernet is specifically used for establishing secured physical network connectivity with local network infrastructure

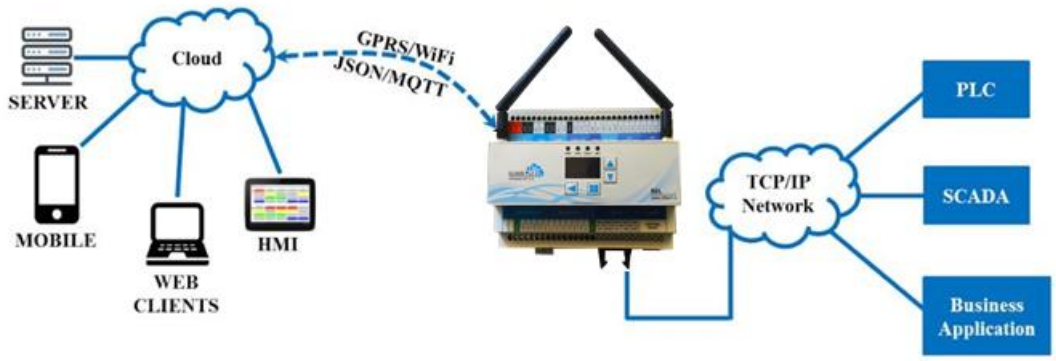
Specification

- W5500 IC with SPI serial interface
- IEEE 802.3 Gigabit Ethernet Compliant
- Communication protocols: TCP/IP, HTTP, FTP, MQTT, UDP,JSON….
- 3.3V operation with 5V I/O signal tolerance
- Low Power Consumption <200mW at 1.25Gbps.

Functional Diagram



Use Case



Example Code

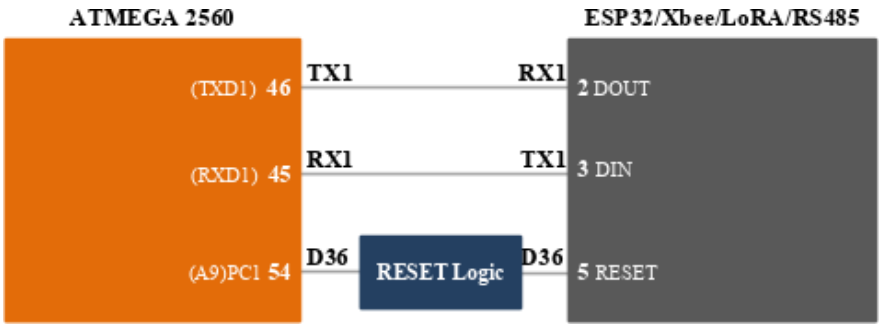
You may look into the following link for examples on Ethernet.
https://www.arduino.cc/en/Reference/Ethernet

8.10. ESP32/XBEE/LoRA/RS485

This is Add-On pluggable module. One among ESP32, XBEE, LoRA or RS485 is comes with the product. For more details on this, look into Order Information Table.

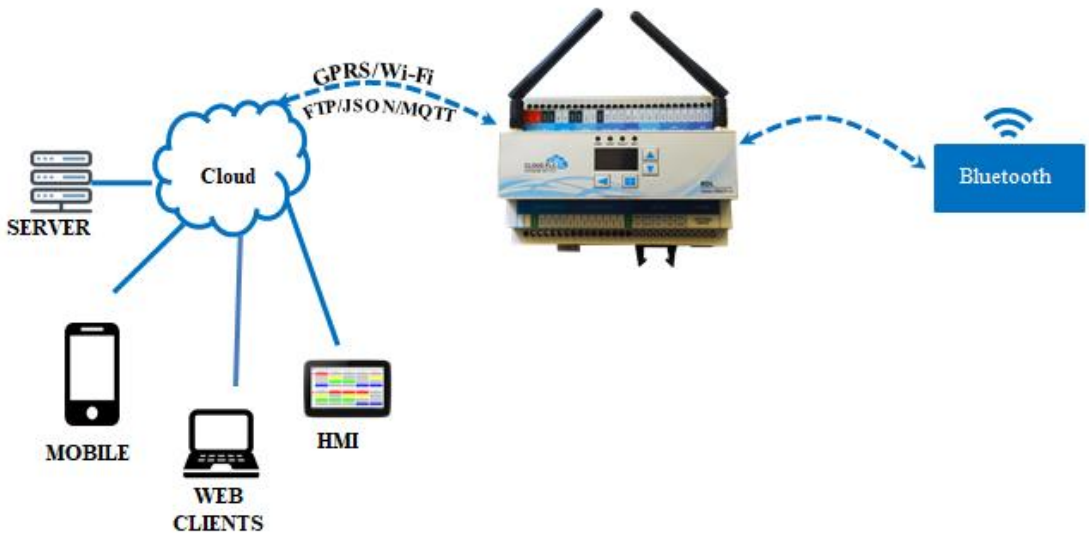
This is specifically used for wireless connectivity with existing infrastructure.

Functional Diagram

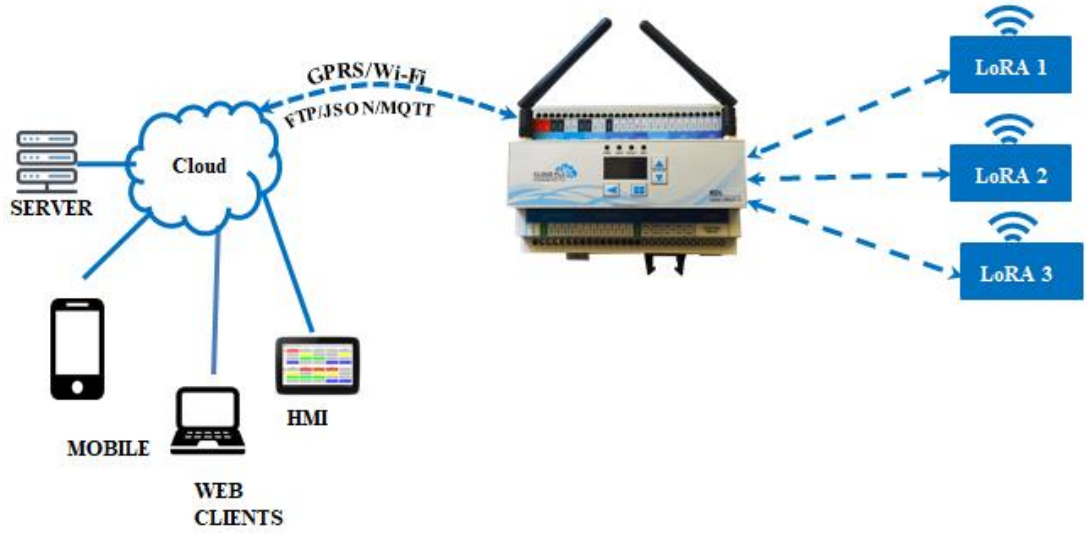


Use Case

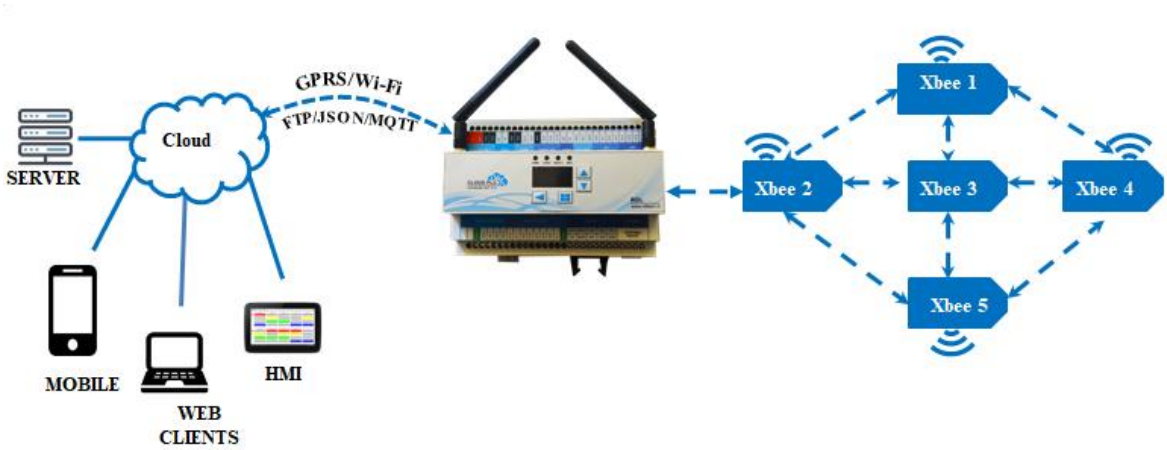
1. Interfacing Industrial Cloud PLC with Bluetooth



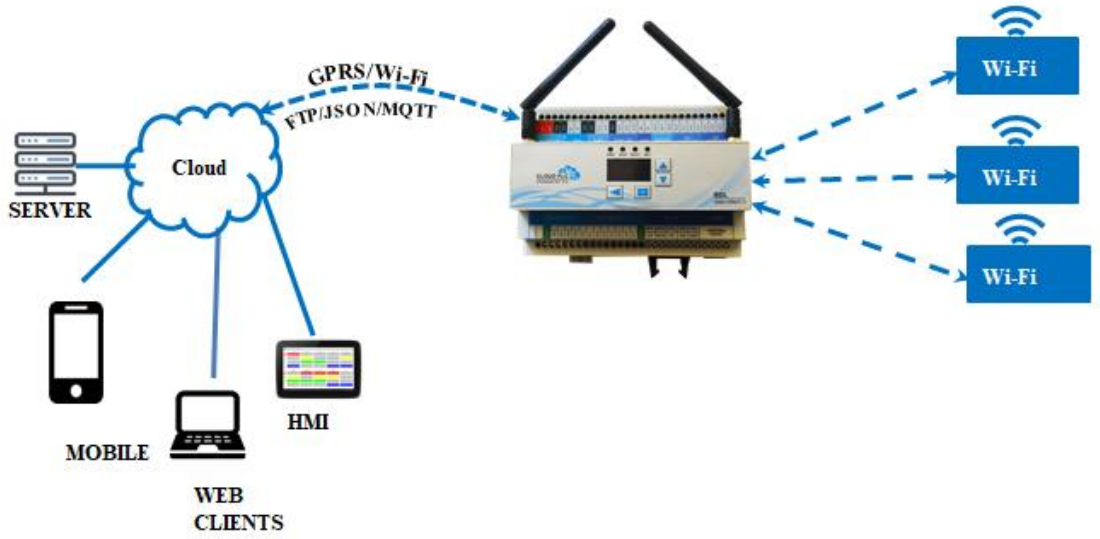
2. Interfacing Industrial Cloud PLC with LoRA



3. Interfacing Industrial Cloud PLC with Xbee



4. Interfacing Industrial Cloud PLC with Wi-Fi (ESP32)



Example Code

You may look into the following link for examples on esp8266.

https://www.arduino.cc/en/Reference/WiFiServer

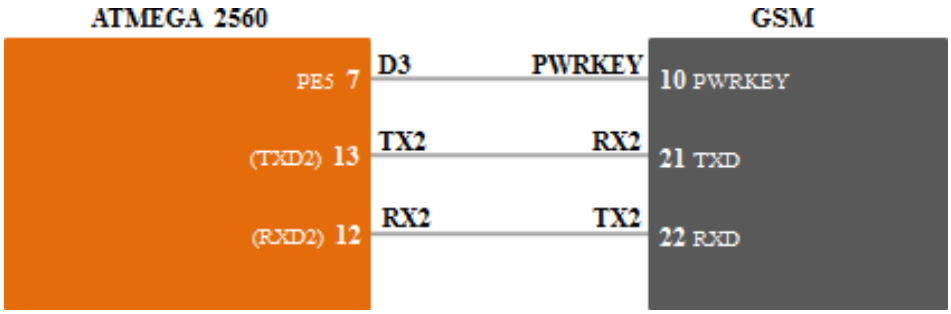
8.11. GSM/GPRS

This is specifically used for M2M and remote data logging and control applications.

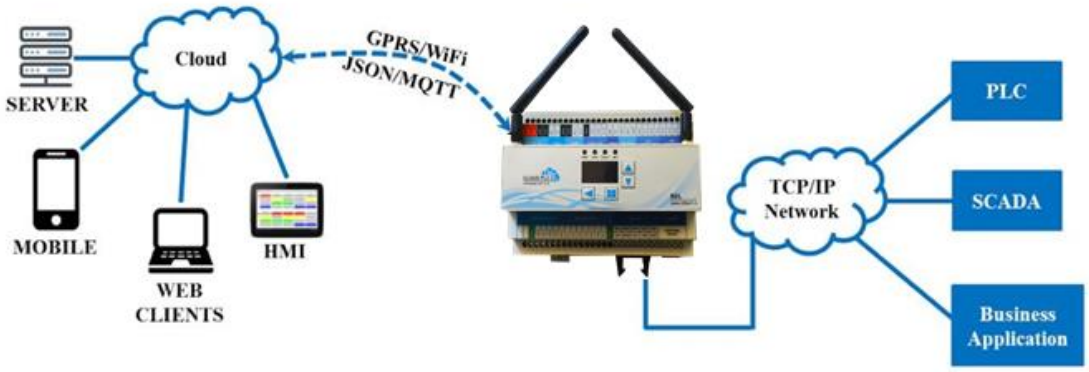
Specification

- Quectel M95, Quad-Band 850/900/1800/1900MHz.
- Serial interface for direct communication with PC or MCU.
- Configurable baud rate.
- Power controlled using 29302WU IC.
- ESD Compliance.
- Enabled with Audio jack.
- With push pull SIM card holder.
- With Stub antenna and SMA connector.

Functional Diagram



Use Case



Example Code

You may look into the following link for examples on GSM.

https://www.arduino.cc/en/Tutorial/GSMExamplesSendSMS

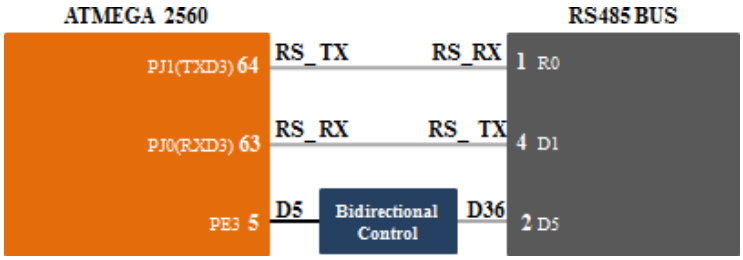
8.12. RS485 Modbus

- Modbus is an Industrial standard serial communication protocol.
- Open protocol
- Information is stored in the Slave device in four different tables.
  Two tables store on/off discrete values (coils) and two store numerical values (registers). The coils and registers each have a read-only table and read-write table.
- Each table has 9999 values.
  Each coil or contact is 1 bit and assigned a data address between 0000-270E.
  Each register is 1 word = 16 bits = 2 bytes and also has data address between 0000 and 270E.
- Supported Functions are
  - Coils
  - Discrete inputs
  - Input Registers
  - Holding Registers.

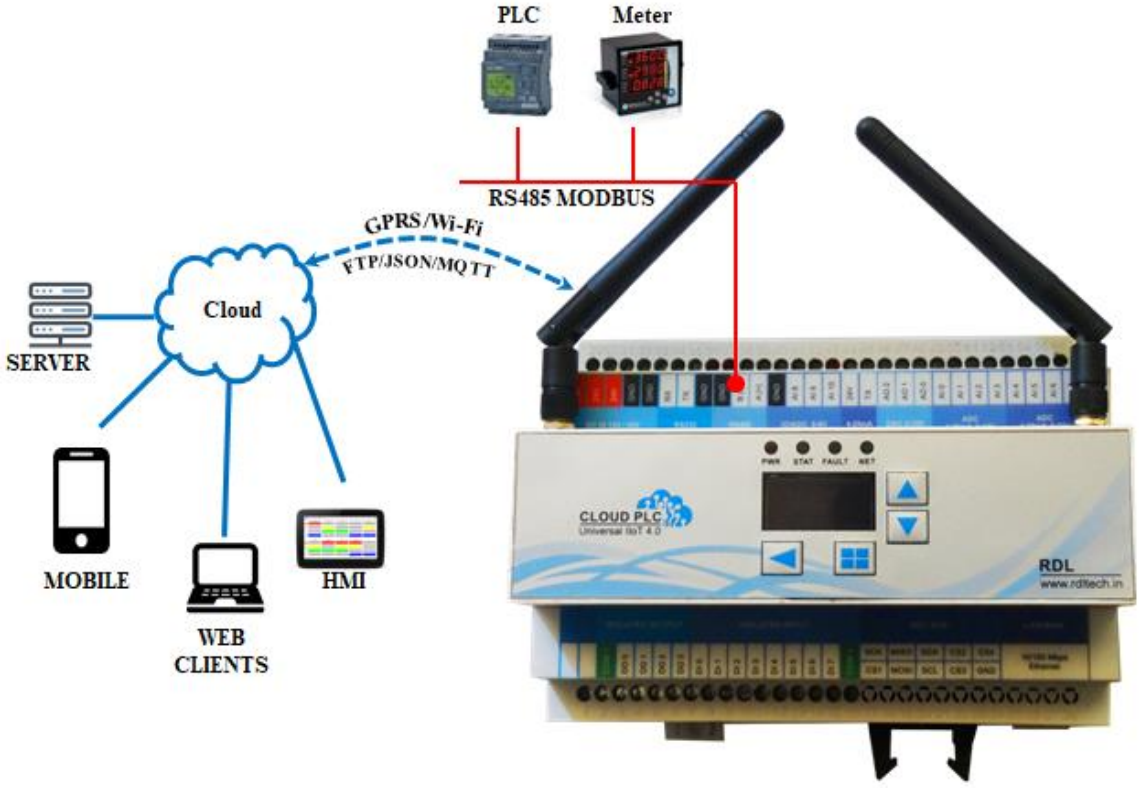
Specification

- LTC485 IC.
- Supports slave address up to 32.
- Supports Modbus protocol with RTU and ASCII formats.
- Configurable baud rate from 4800 to 115200.
- Configurable packet format (data bits, parity bit, stop bits)

Functional Diagram



Use Case



Example Code

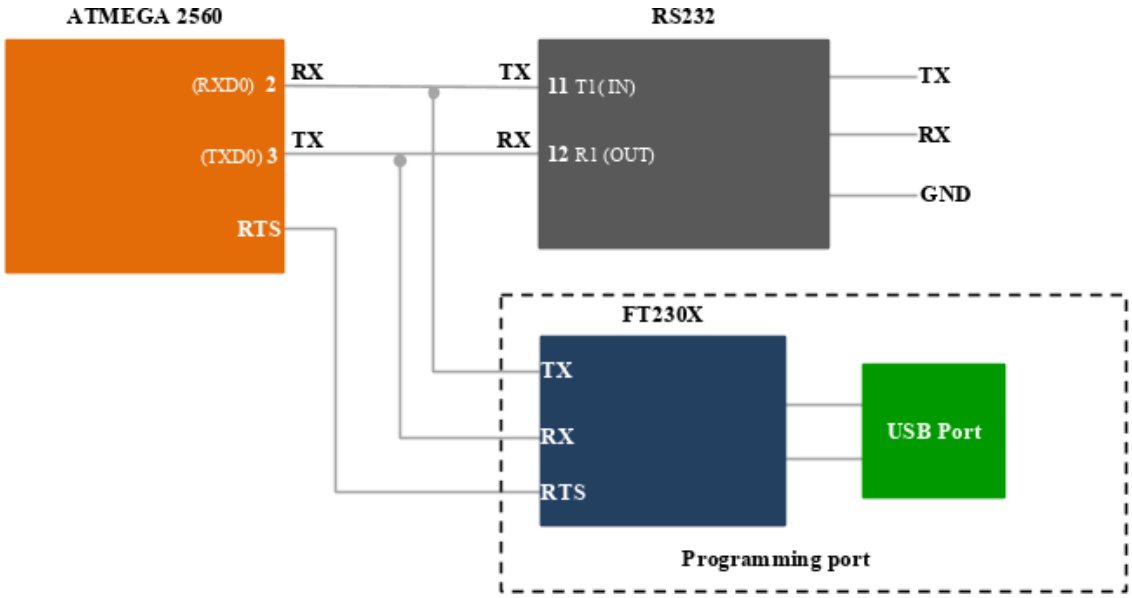
You may look into the following link for examples on Modbus examples.

https://playground.arduino.cc/Code/ModbusMaster

8.13. RS232/FT232/Program

Used for programming the board. When in user mode, the port could be used for data communication.

Functional Diagram



ATTENTION: When programming the board, it is recommended to remove any connection made to the RS232 serial pins in order to ensure proper functioning of the system.

Example Code

You may look into the following link for examples on FT232/MAX232 serial communication.

https://www.arduino.cc/reference/en/language/functions/communication/serial/

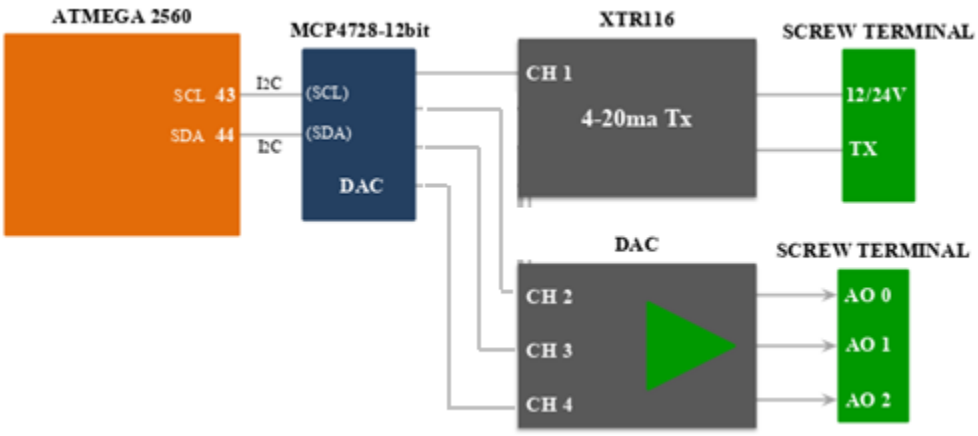
https://www.arduino.cc/en/Tutorial/SoftwareSerialExample

8.14. DAC

Specification

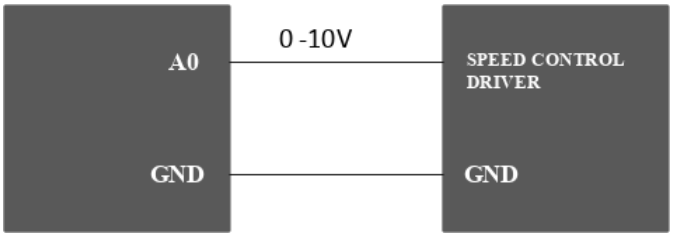
- MCP4768 with I2C serial interface
- Quad, 12-bit voltage output
- Channel: 4 (buffered outputs)
- Internal Voltage Reference
- Output Voltage Range using 0-10V

Functional Diagram

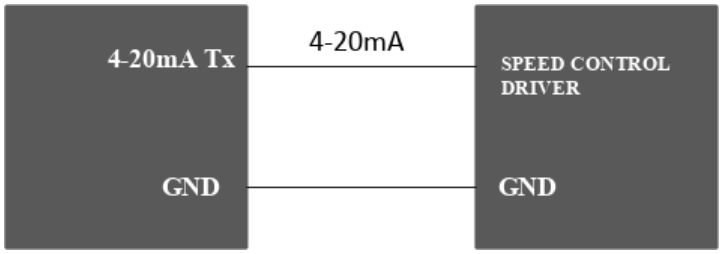


Application Wiring

1. Interfacing DAC with Motor Speed Control Module

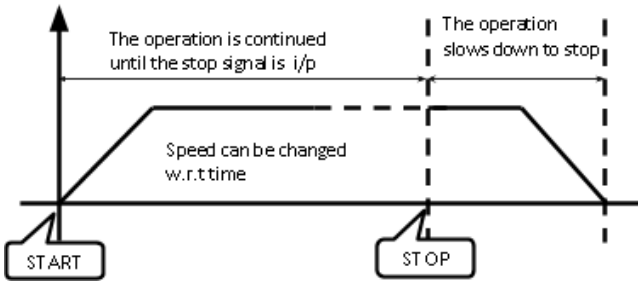


2. Interfacing DAC with Motor Speed Control Module using Loop Current

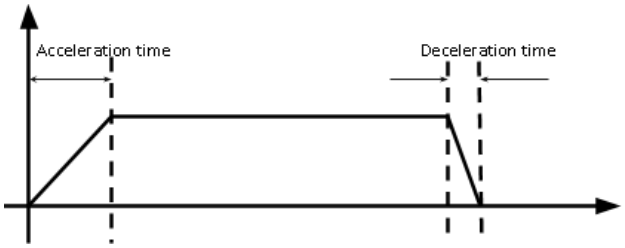


Use Case

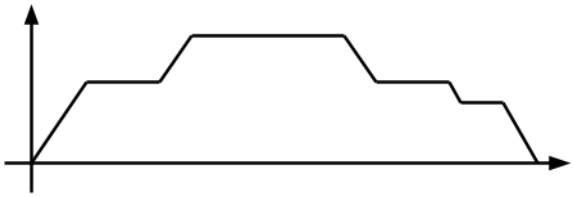
1. Motion Control



2. Acceleration and Deceleration



3. Jog Operation and Trapezoidal Control Operation



The speed can be freely changed until the operation starts to decelerate to stop

Example Code

You may look into the following link for the Arduino library and example on DAC.

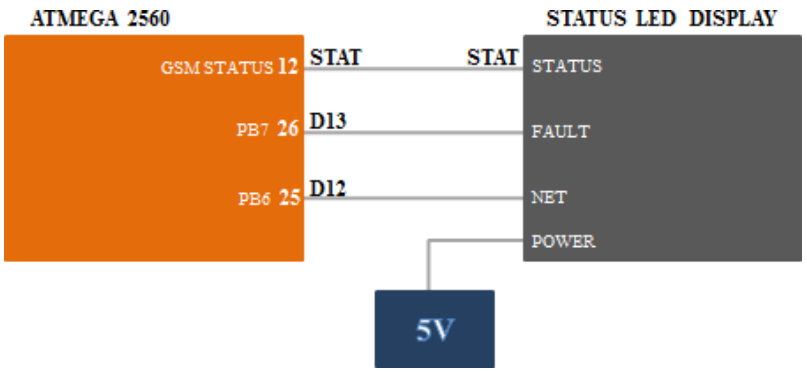
https://github.com/hideakitai/MCP4728

8.15. Status LED Display

Status LED's can be programmed as per used needs for visual indication of an event.

Refer Digital Output Section

Functional Diagram

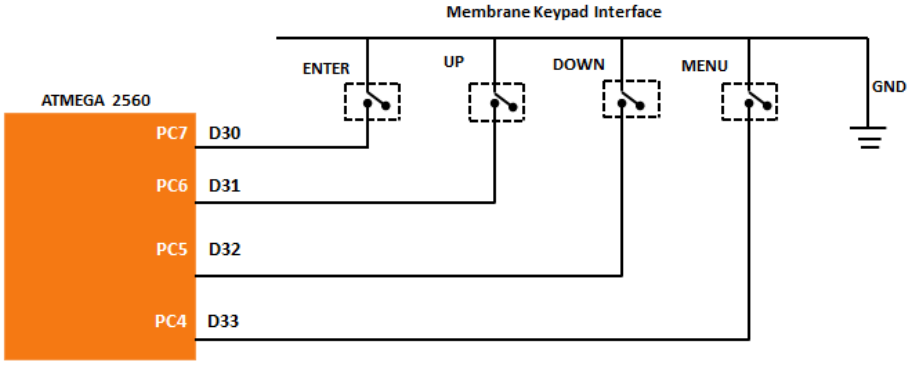


Example Code

You may look into the following link for more details on programming LED pins.

https://www.arduino.cc/en/Reference.digitalWrite

8.16. Membrane Keypad



Example Code

You may look into the following link for example on reading a digital pin.

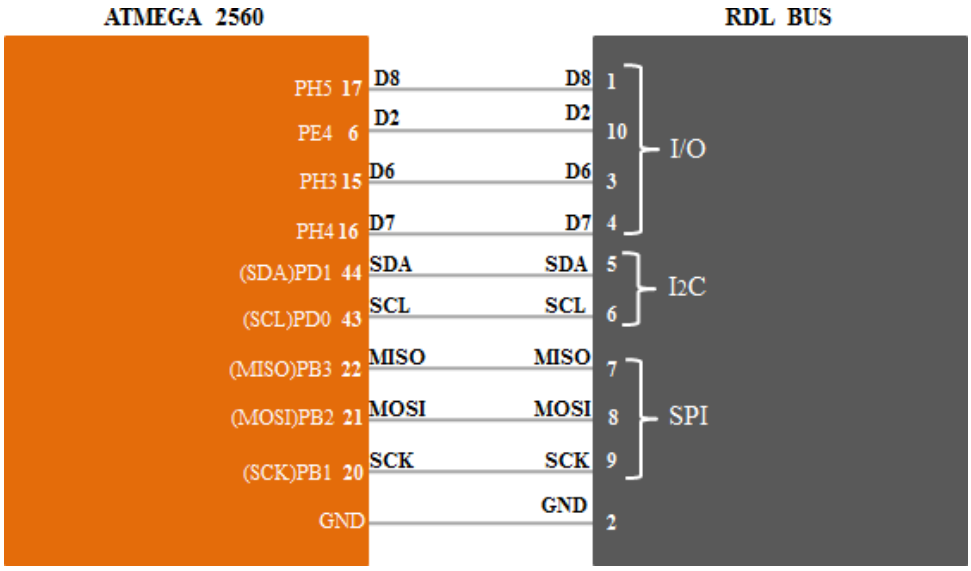
https://www.arduino.cc/reference/en/language/functions/digital-io/digitalread/

8.17. RDL Bus

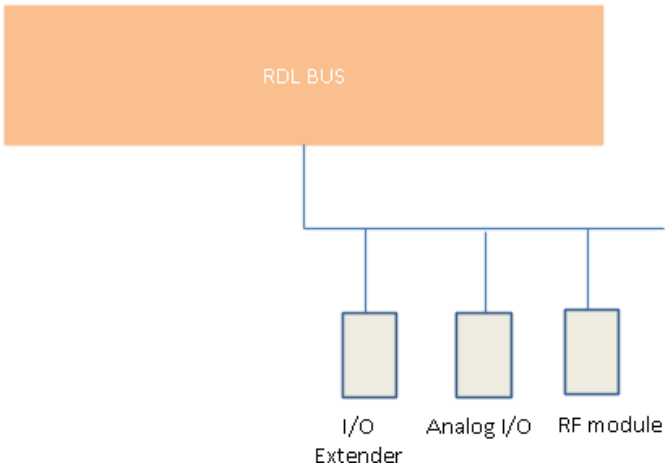
Specification

- Extend I/O pins for communicating with external devices.
- Extends SPI pins, I2C pins, UART pins and Digital I/O pins.

Functional Diagram



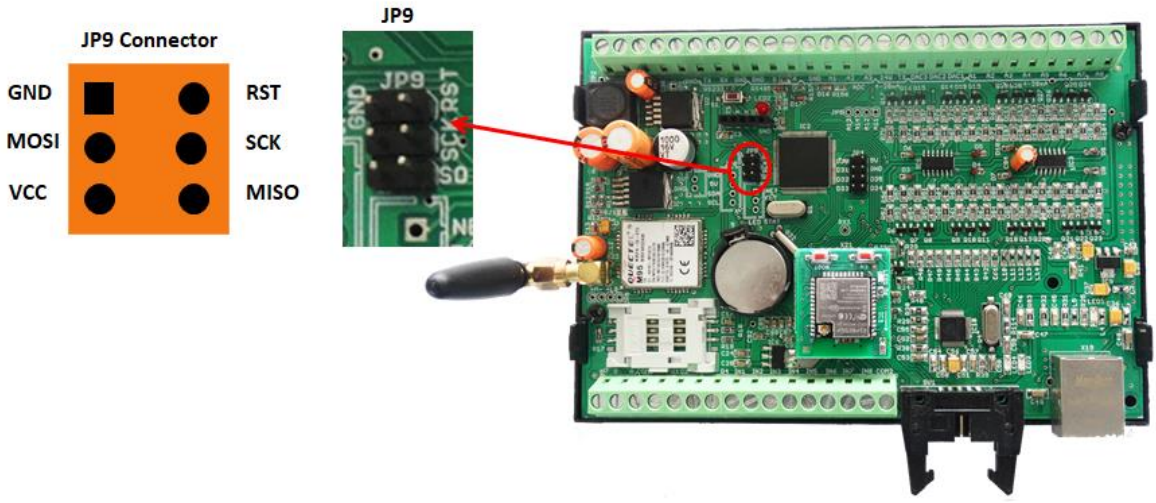
Application Wiring



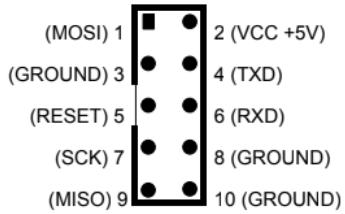
9. Boot Load Instructions:

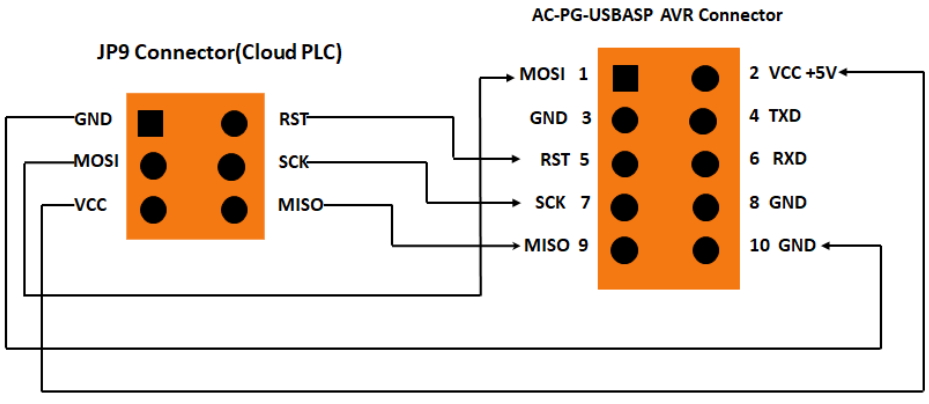
Cloud PLC supports multiple open source IDE for programming , you can choose any one of the below given method to build the custom programming/solution.

9.1 Using USBASP AVR Programmer



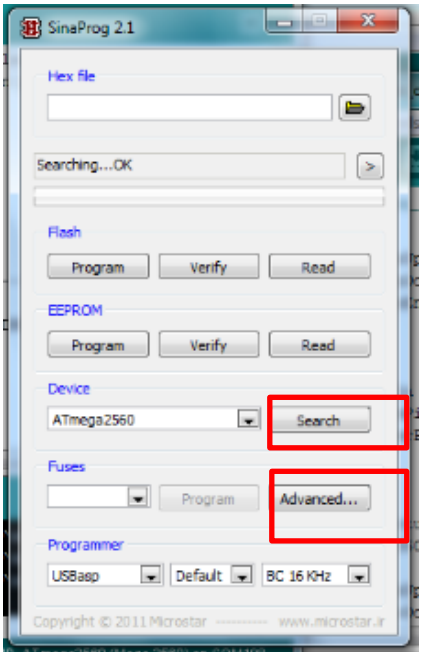
AC-PG-USBASP AVR Programmer



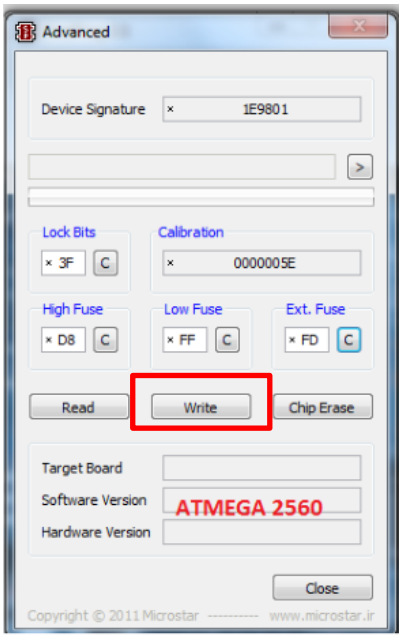


Burn Bootloader into Cloud PLC

STEP 1: Open SinaProg 2.1 Software, Click on Search command, you get searching OK



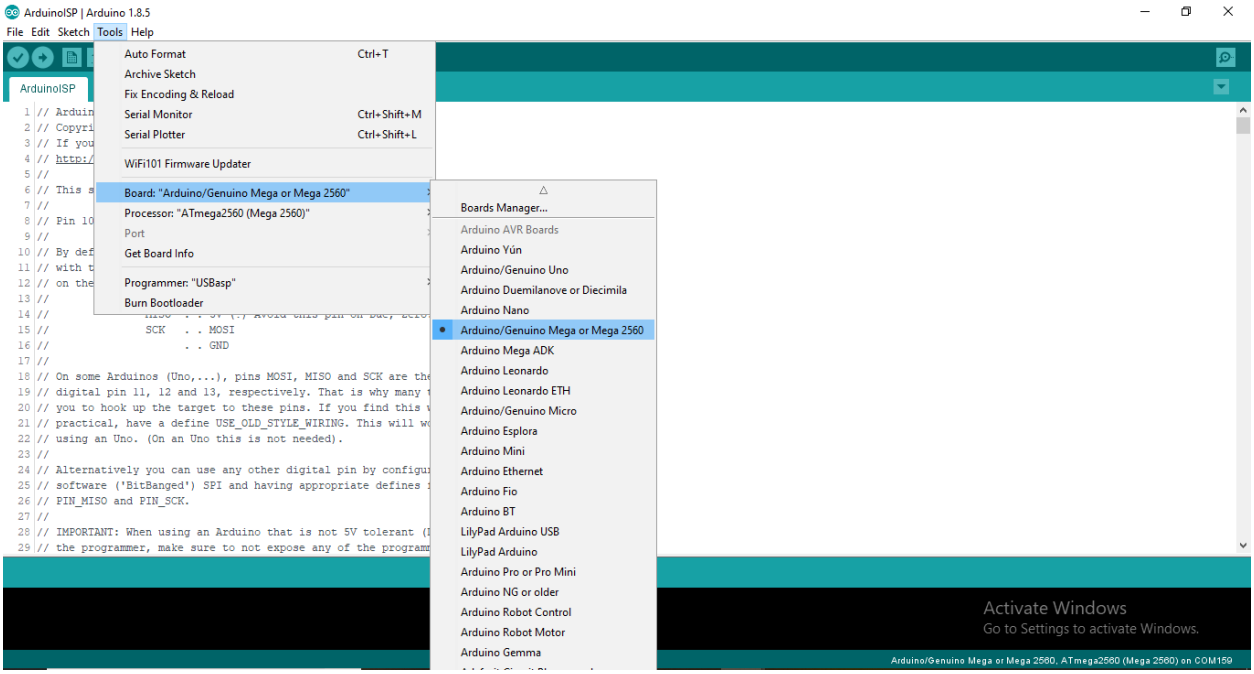
STEP 2: Click on **Advanced** to set the fuse bits Lock bits **3F** High Fuse **D8** Low Fuse **FF** Ext Fuse **FD**



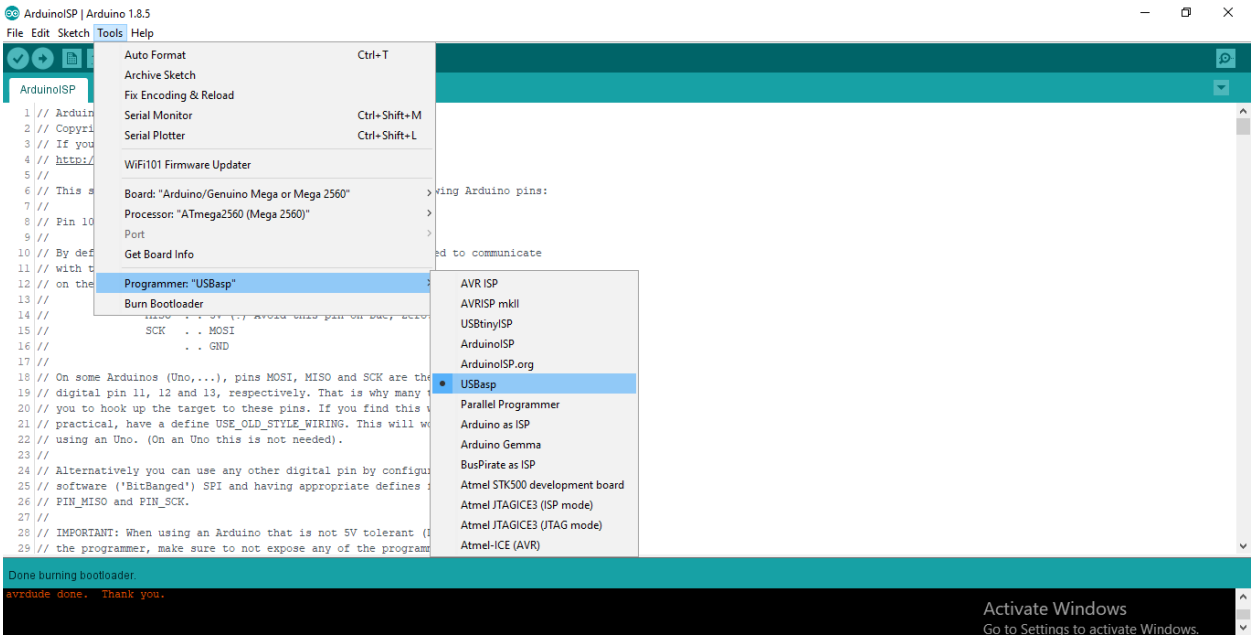
STEP 3: Click on **Write**

STEP 4: Go to Tools > Board and choose "Arduino/Genuino Mega or Mega2560 " as the

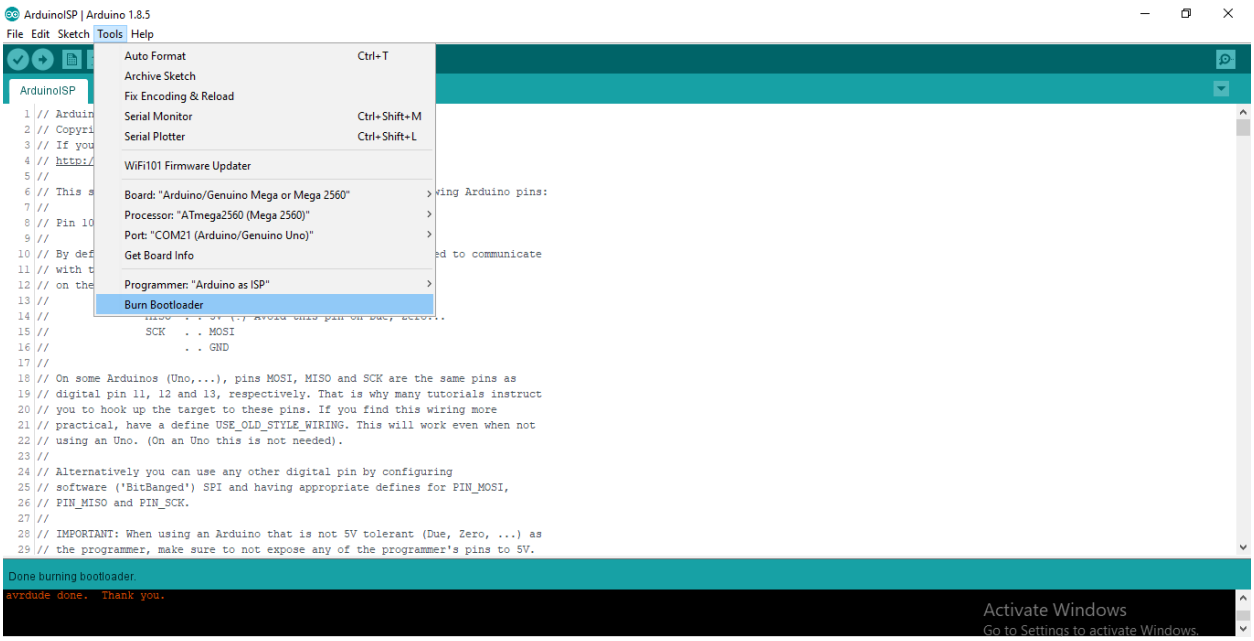
board of the target



STEP 5: firstly we select "USBasp" from the Programmer

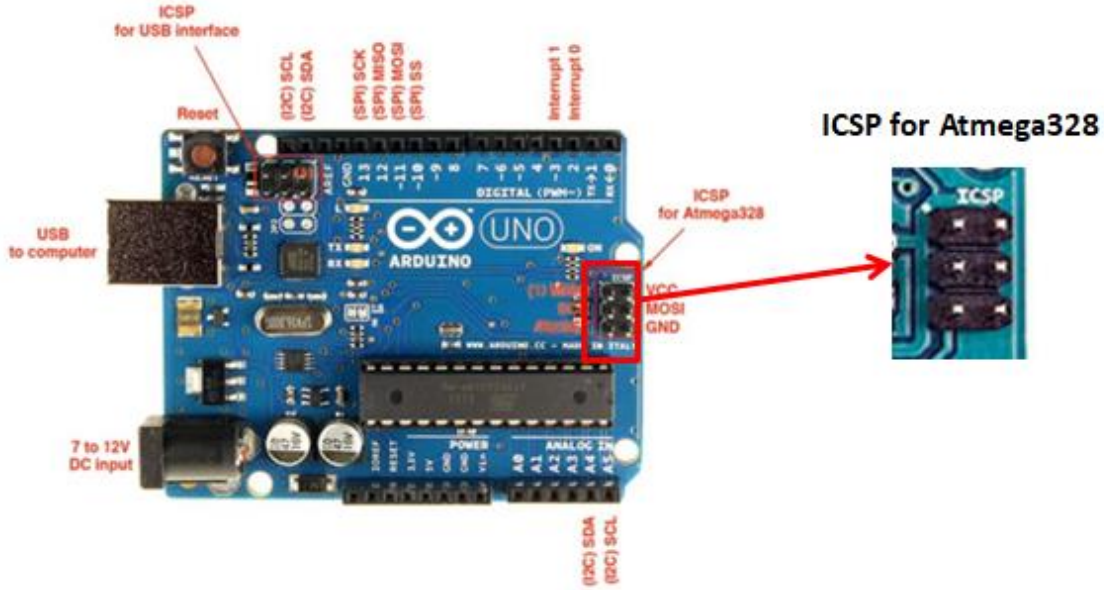


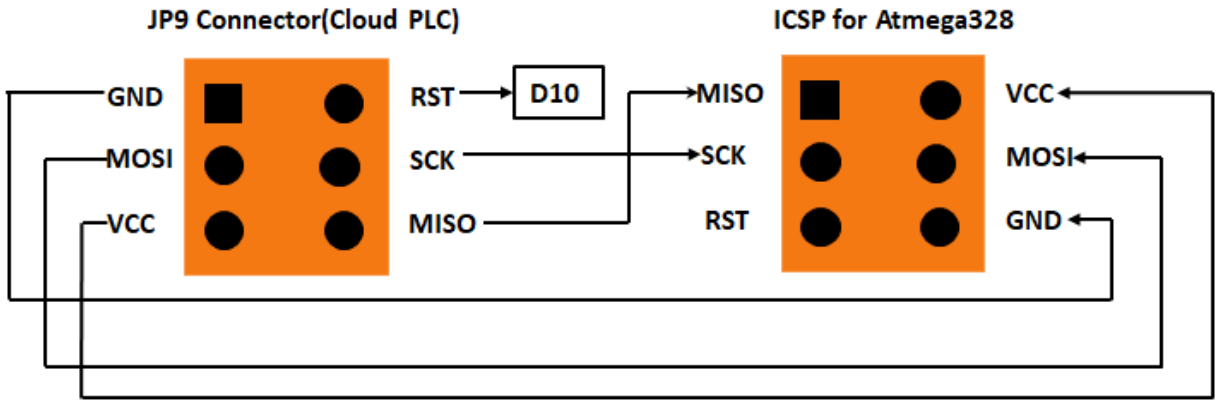
STEP 6: After that, click on "Burn Bootloader" from the Tools menu



Arduino IDE will display "Done burning bootloader" when it is done.

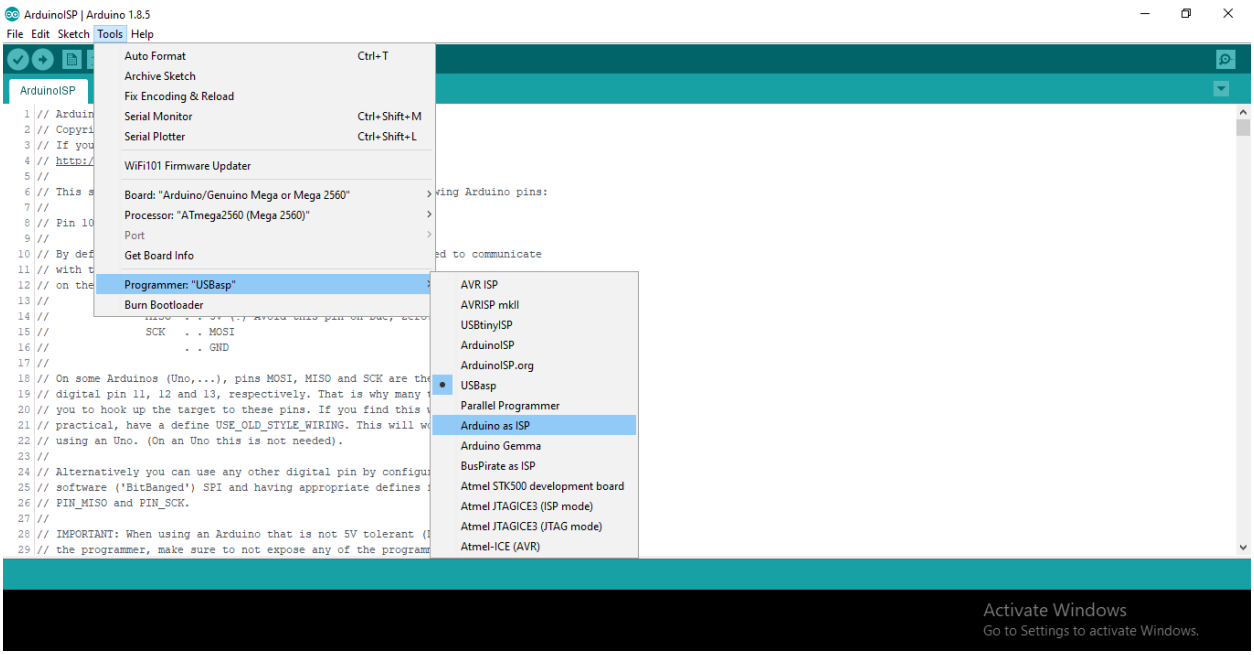
9.2 Using Atmega328



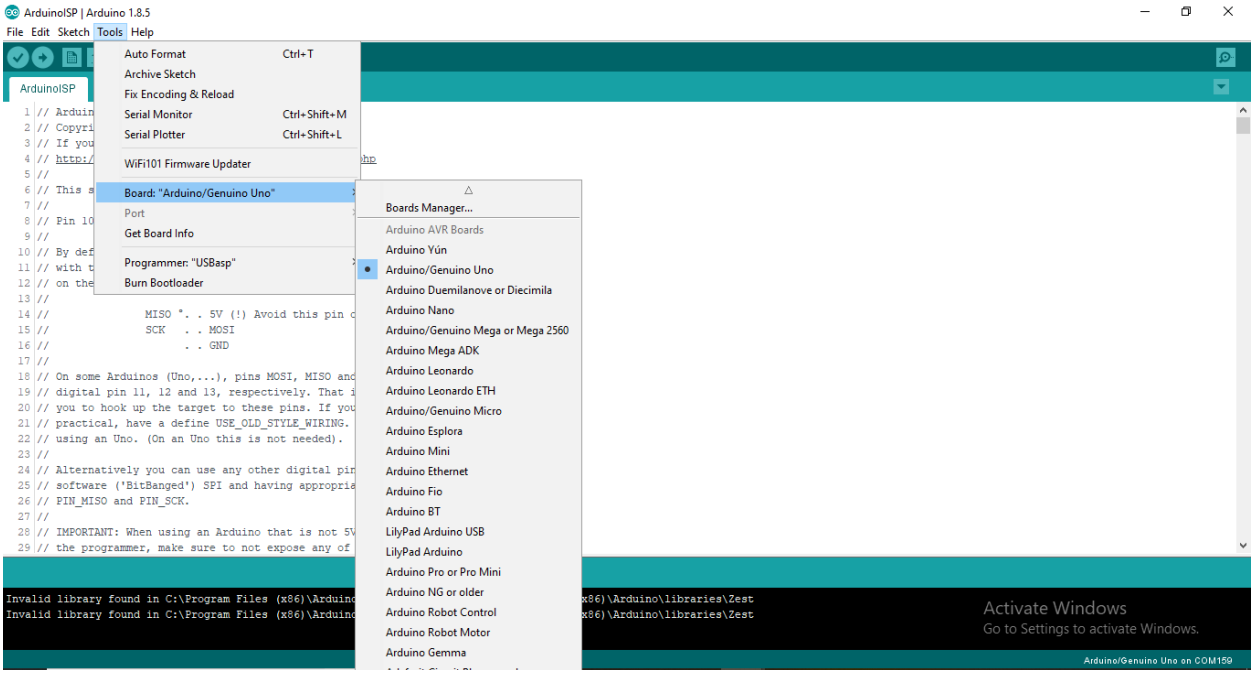


Burn Bootloader into Cloud PLC

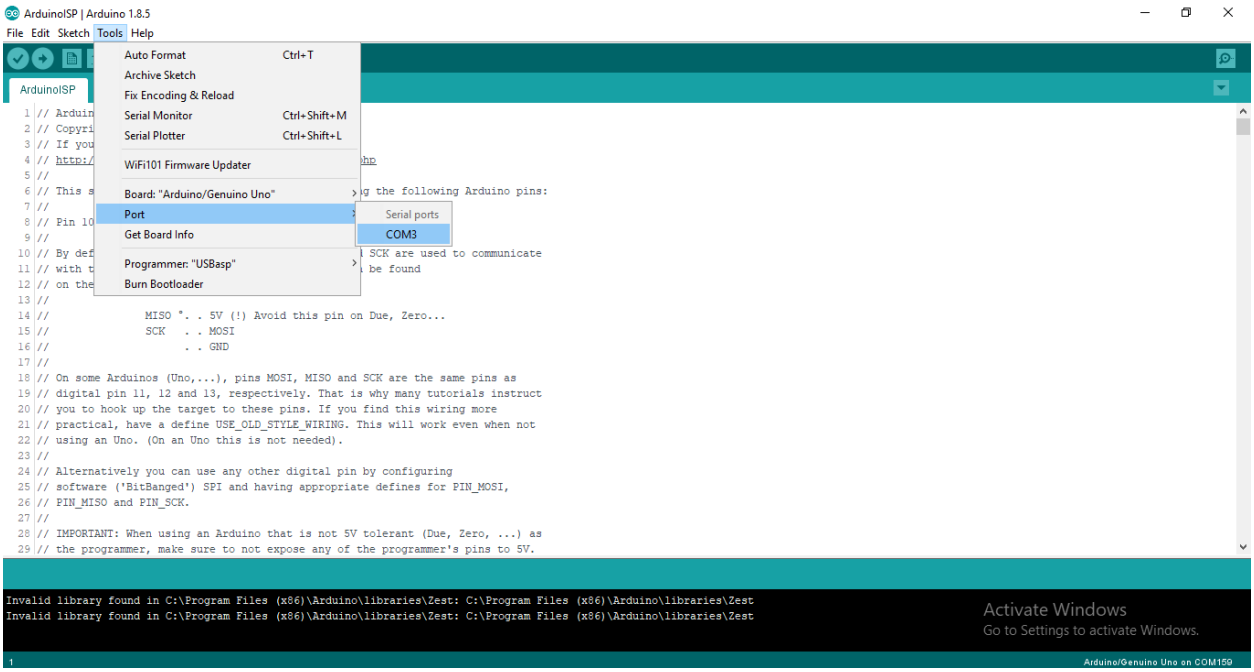
STEP 1: An ISP (In-system programming) programmer is needed to burn bootloader. In this we will use a Cloud PLC as an ISP programmer. To prepare Cloud PLC as ISP programmer, firstly we select "ArduinoISP" from the File > Examples menu.



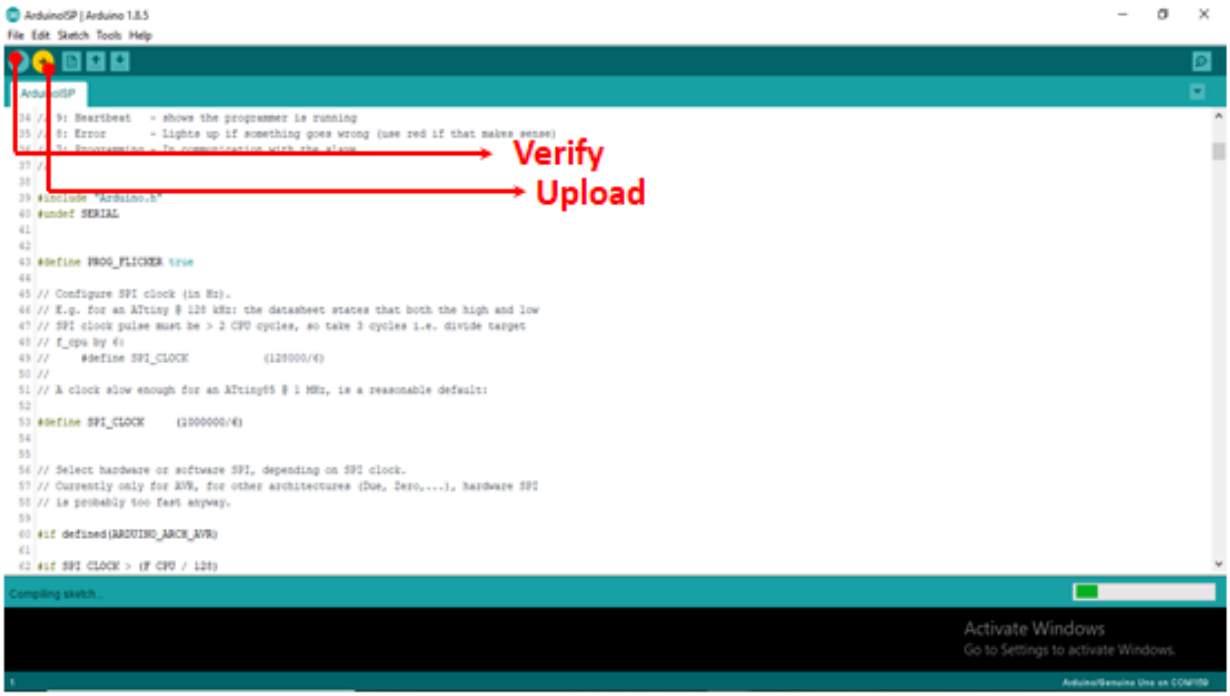
STEP 2: Select the Board **Arduino/GenuinoUno**



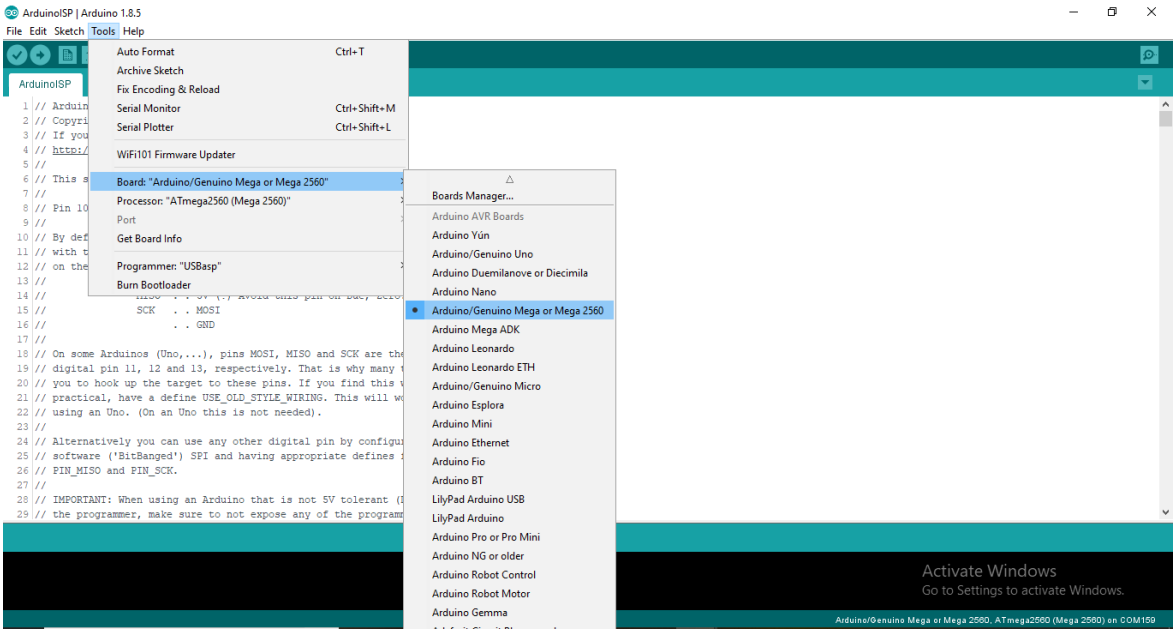
STEP 3: the **Port**



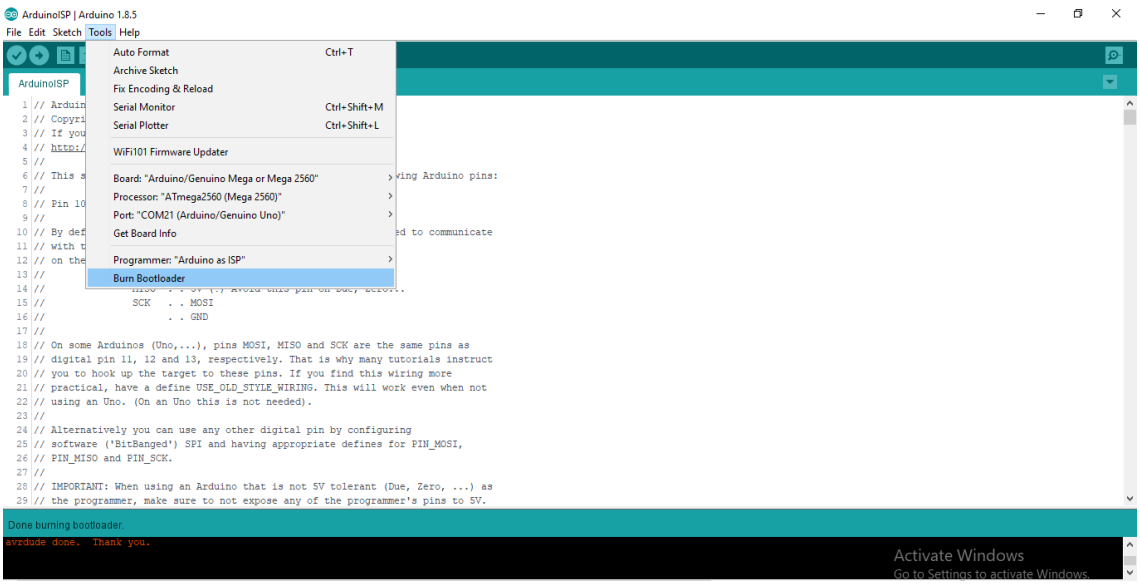
STEP 4: Verify and upload the code



STEP 5: Once the code is uploaded, do the wiring up, go to Tools > Board and choose "Arduino/Genuino Mega or Mega2560 " as the board of the target .



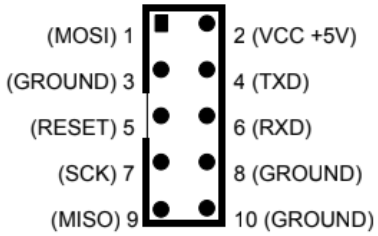
STEP 6: After that, click on "Burn Bootloader" from the Tools menu, the ISP programmer will start to burn the bootloader into the target Arduino. It usually takes few minute to complete burning bootloader.

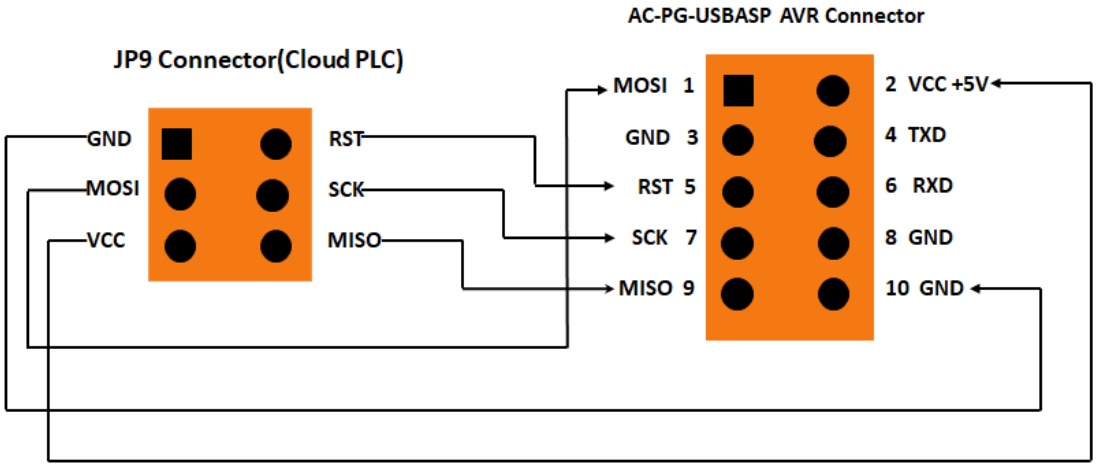


STEP 7: Arduino IDE will display "Done burning bootloader" when it is done.

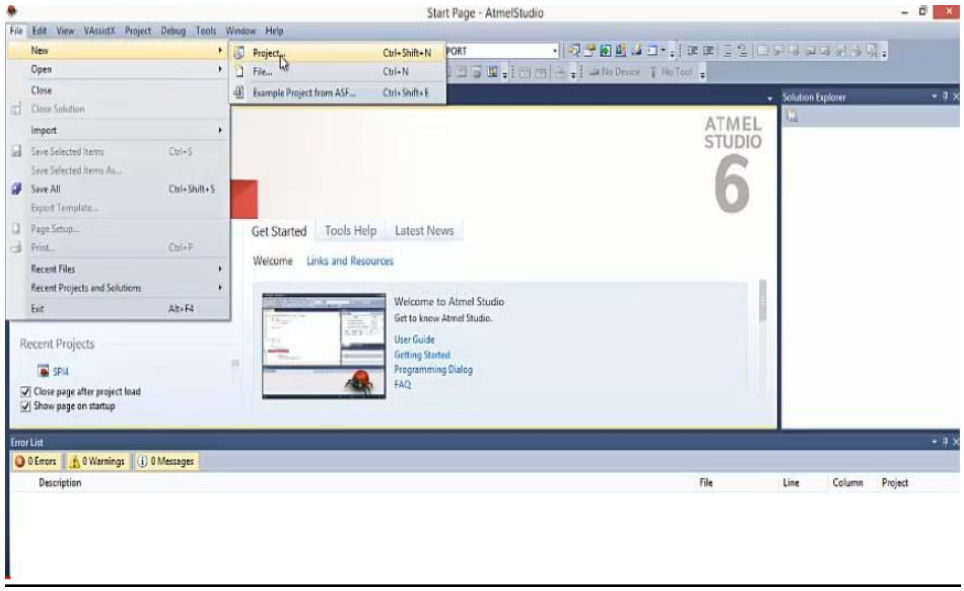
9.3 Writing the Code-Atmel Studio

AC-PG-USBASP AVR Programmer

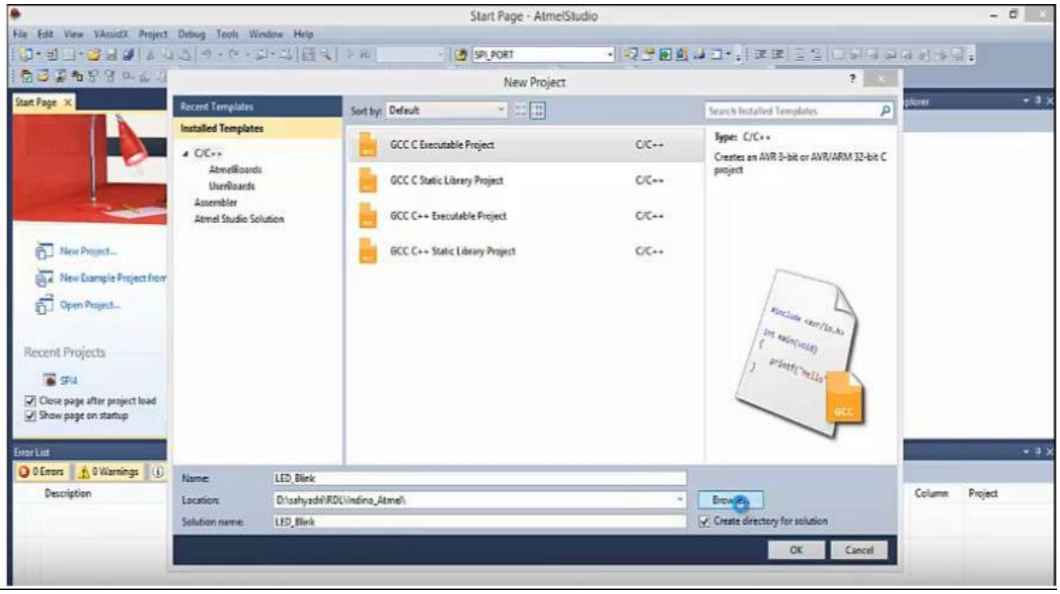




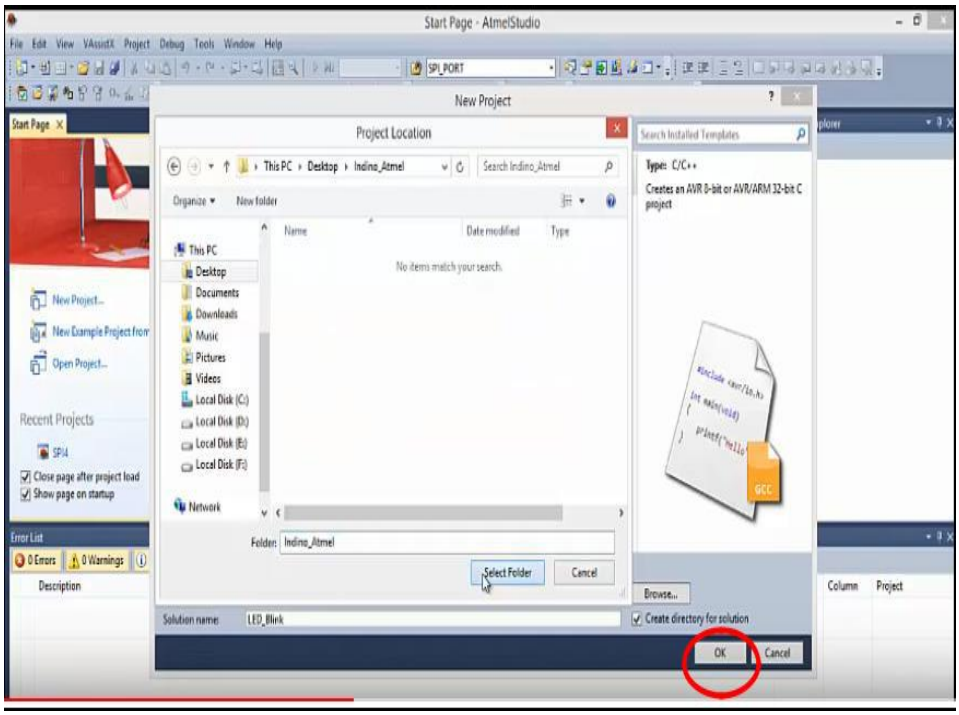
Step 1: Open Atmel Studio, go to file → new → project.



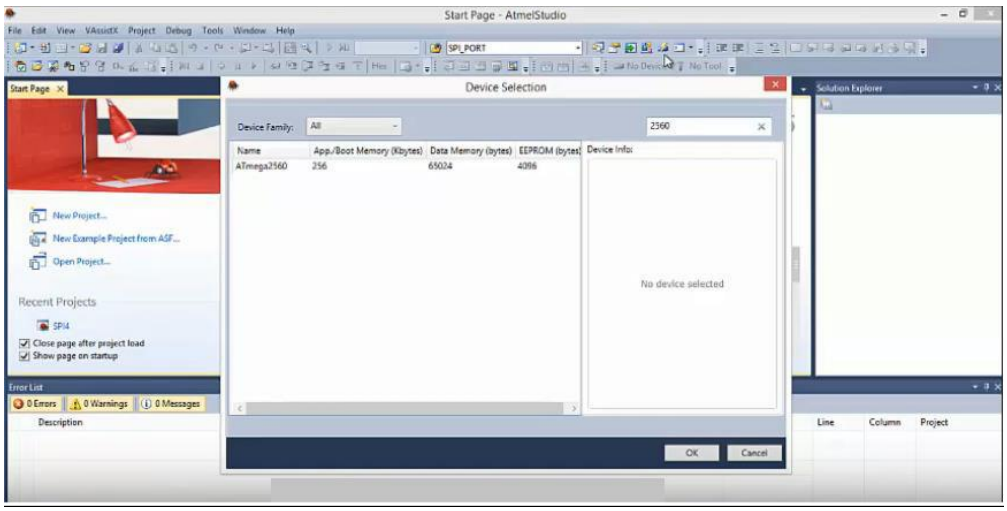
Step 2: In new project, type the desired file name and click on browse.



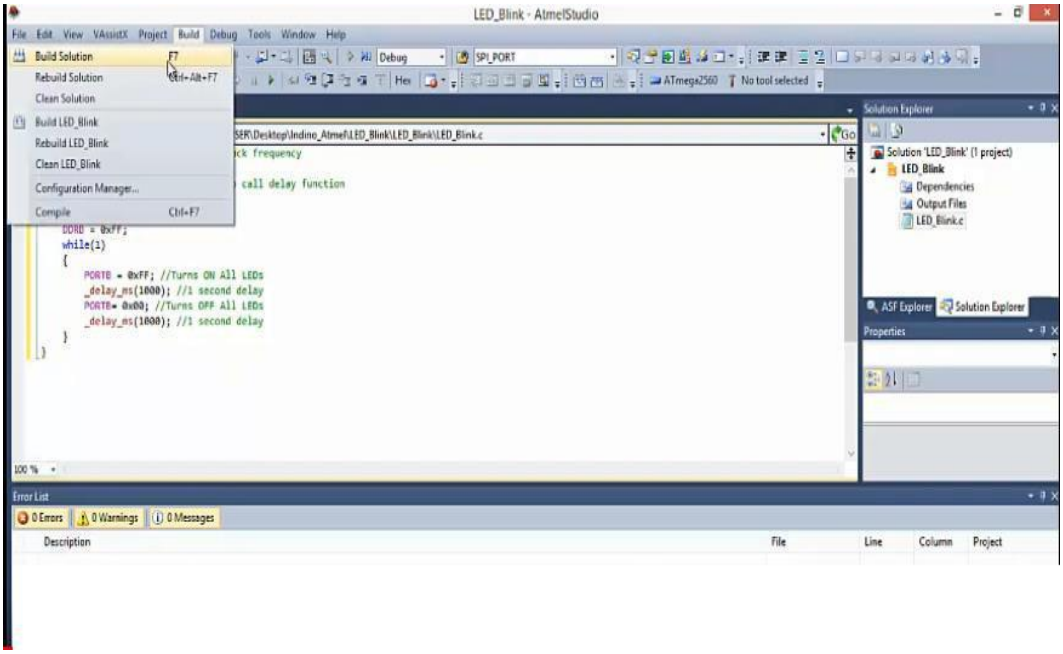
Step 3: Go to desktop → Create new folder → Select Folder, press OK.



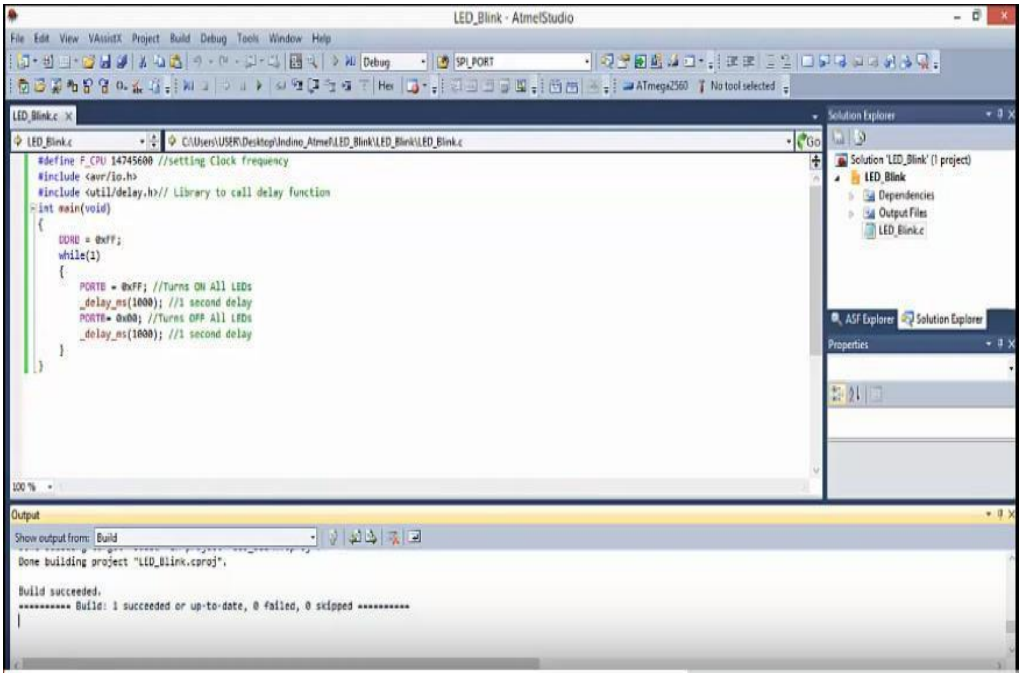
Step 4: Select the device as ATmega 2560 and click on OK.



Step 5: Copy and paste the program and save, go to build → build solution, to check for errors in program



Step 6: If your program has no errors it will show the build window as shown below.



10. Power Supply

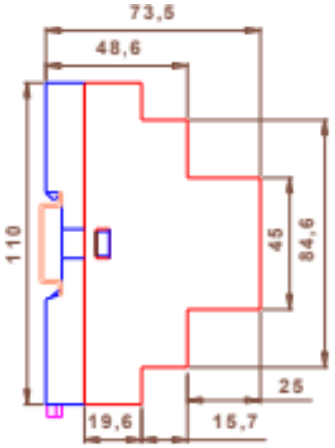
Model	Input Voltage	Vmin	Vmax
7000 - 7009	12V - 36V	12V	30V

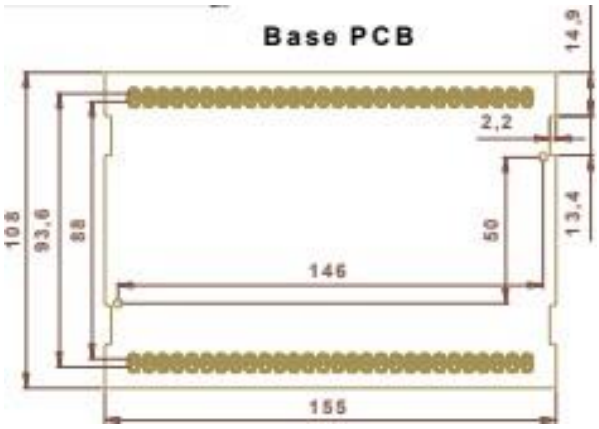
ATTENTION: Recommended to use Meanwell power supplies of 24V 2A

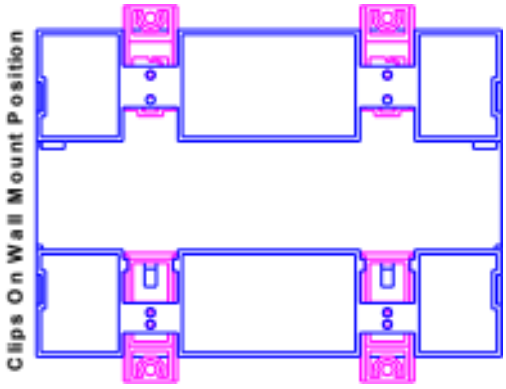
11. Order Information Table

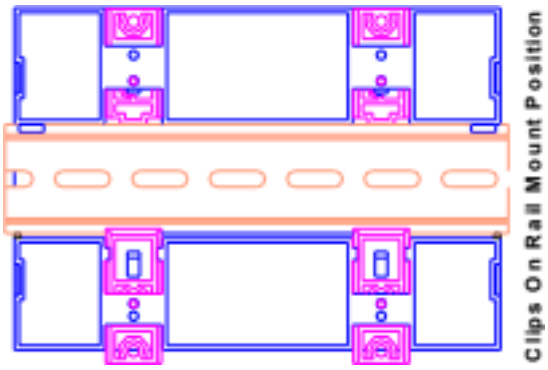
Model	RDL9000	RDL9001	RDL9002	RDL9003	RDL9004	RDL9005	RDL9006	RDL9007	RDL9008	RDL9009
Digital Input(DI)	X	X	X	8	8	8	8	8	8	8
Digital Output(DO)	X	X	X	4	4	4	4	4	4	4
Analog 0-10V / 4-20MA	X	X	X	X	X	8	8	8	8	8
GPRS	1	X	X	1	X	X	1	1	1	1
ETHERNET 10/100MBPS	X	1	X	1	1	1	1	X	X	1
WIFI / Bluetooth	X	X	1	X	1	1	X	1	X	1
ZigBee/LoRA	X	X	X	X	X	X	X	X	1	X
RS485	1	1	1	1	1	1	1	1	1	1
RS232	1	1	1	1	1	1	1	1	1	1
SD Card	1	1	1	1	1	1	1	1	1	1
RTC	1	1	1	1	1	1	1	1	1	1
DAC/4-20MA TX	X	X	X	X	X	X	X	X	X	1

12. Mounting and Mechanical Dimensions









13. References and Datasheets

1. http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf
2. http://www.ti.com/lit/ds/symlink/max232.pdf
3. http://ww1.microchip.com/downloads/en/DeviceDoc/40001441F.pdf
4. http://ww1.microchip.com/downloads/en/DeviceDoc/20005685A.pdf
5. http://www.ti.com/lit/ds/symlink/lm2576hv.pdf
6. https://www.vishay.com/docs/83513/tcmd1000.pdf
7. http://www.analog.com/media/en/technical-documentation/data-sheets/485fm.pdf
8. http://www.ti.com/lit/ds/symlink/lm317.pdf
9. http://www.ti.com/lit/ds/symlink/ads1115.pdf
10. http://ww1.microchip.com/downloads/en/DeviceDoc/20005045C.pdf
11. https://www.fujitsu.com/uk/Images/MB85RC256V-20171207.pdf
12. http://ww1.microchip.com/downloads/en/DeviceDoc/mic811.pdf
13. http://wizwiki.net/wiki/lib/exe/fetch.php?media=products:w5500:w5500_ds_v106e_141230.pdf
14. http://www.analog.com/media/en/technical-documentation/data-sheets/485fm.pdf
15. http://ww1.microchip.com/downloads/en/DeviceDoc/22187E.pdf
16. http://www.ti.com/lit/ds/symlink/xtr115.pdf