

**Generic Library  
for  
Microcontroller  
as  
Modbus RTU Slave**

# Why to implement Modbus RTU in Microcontroller?

- Cheaper solution for deploying **DCS** (Distributed Control System)
- The data transaction can be implemented over **IoT** using devices like **ESP-32**
- Large number of I/Os can be handled efficiently
- Sensors integrated with peripherals of Microcontroller can add more functionality for Automation
- Modbus RTU provides longer distance range compared to UART, I2C, SPI etc.

# What is this project about?

- Create a **Generic C Library** of **Modbus RTU Slave** implementation that can be integrated to any existing microcontroller
- The Library has already been developed and tested successfully on **Cypress PSoC5** based Development Kit and **TI Stellaris Launchpad** Development Kit
- To take this further and considering **Arduino** is a widely used platform for both educational and industrial purpose, The same library is tested successfully, first on **Arduino Mega** and then on **Arduino Uno**
- This project is about openly sharing this Library with fellow developers, no permission required for them to ask to use it commercially

# Why Arduino Mega and not Uno?

- If this project is to be carried forward for any industrial application, considering the complexity and volume of **sensor integration, I/O, ADC, other Communication peripherals, Size of EEPROM, FLASH Size**, etc., **Arduino Mega** is preferred, although as the library works with **Arduino Uno**, the choice is up to the developer.
- The library can be tested on other Arduino Boards as well. Feedbacks of them are also welcome.

# How to use this library?

- The **modbus\_rtu.c** and **modbus\_rtu.h** are the library files. Add them to your existing microcontroller project. Changes can be added in **main.c** file. Minimum **512 Bytes** of **EEPROM** are required to test this library.
- Get an **RS-485 to UART Converter** for field test. If just want see the functionality, use **qModbus** PC Tool and a **USB to UART Converter**.
- Set up **UART** on **Baud Rate 9600** (Changeable) and **Data Receive Interrupt**. Define **UART Transmit** function according to the datasheet of microcontroller.
- Set up a **timer of 1 ms**, with a functionality to start and stop it when required.
- Refer **ard\_mrtus.ino**, try to apply the same philosophy of code and contact us at [\*\*philomath@embverse.com\*\*](mailto:philomath@embverse.com) if any assistance required.