**Microcontroller theory**

Microcontroller unit (MCU) is a small size, special purpose computer. It is small enough in order to be integrated on a small circuit in which will do specified tasks or applications. MCU itself comes with memory, input, output peripherals and processor. Program to run the MCU is stored in Read-only memory (ROM) and usually not change in production. A microcontroller is usually designed to run in small size and at low cost, which is compatible to be embedded in other system in order to control actions of the system automatically.

**Types**

- Intel 8051

- STMicroelectronics STM8S (8-bit), ST10 (16-bit) và STM32 (32-bit)

- Atmel AVR (8-bit), AVR32 (32-bit), và AT91SAM (32-bit)

- Freescale ColdFire (32-bit) và S08 (8-bit)

- PIC (8-bit PIC16, PIC18, 16-bit dsPIC33 / PIC24)

- Renesas Electronics: RL78 16-bit MCU; RX 32-bit MCU; SuperH; V850 32-

bit MCU; H8; R8C 16-bit MCU

- PSoC (Programmable System-on-Chip)

- Texas Instruments Microcontrollers MSP430 (16-bit), C2000 (32-bit), và

Stellaris (32-bit)

**RS-485**

**Introduction**

Nowadays, there are various communication protocols can be used for the thesis, namely I2C, ISP, RS232, RS-485, Bluetooth or Wi-Fi. Each protocol is designed to be suitable for specified purpose with different advantages or disadvantages, which means a perfect protocol does not exist. When making a decision to choose suitable protocols for the thesis, the author had to think about the trade-off between the stabilization and the speed of the communication protocol.

RS-485 is chosen as the main way for components in the system to communicate with each other. RS-485 is defined in 1983 not as a protocol but an electrical interface standard and only specifies the drivers and receivers’ characteristics. It is developed in order to make data rate and transmitting distance are inversely proportional. For instance, the data transmitting speed can reach 10 Mbps within distance of 16 meters or if the distance is extended to 1220 meters, the data rate is lower to 100 kbps. The advantage of RS-485 over RS232, which is developed in 1960, is multiple nodes can be parallel connected to a bus. Additionally, the network can be extended in length and number of nodes easily by using simple connectors. Besides, Wi-Fi, Bluetooth, GSM and MQTT are also implemented in the thesis in order to take the advantages of different communication protocols in different circumstances.

**RS-485**

**Overview**

Table~\ref{table:RS-485HighLightSpecs} shows the remarkable specifications of RS-485. With these characteristics, RS-485 was a robust interface standard and was able to meet the requirements in industries, in which implemented applications that need a stable, fast and reliable connection. Figure~\ref{fig:fullHalfDuplex} demonstrates two ways to implement the connection with RS-485, which are full-duplex and half-duplex. Full-duplex implementations require four-wire (two signal pairs) instead of two-wire in half-duplex implementations; But despite the downside of two-wire implementation is it is limited to half-duplex and needs attention to turn-around delay, in practical applications, half-duplex is most chosen. The reason is full-duplex solution depends on master-slave model, which means the slaves cannot communicate with each other. In modern designs of transceiver, the allowed number of nodes can connect to the bus is up to hundreds.

%RS-485 highlight specs table

\begin{table}[h!]

\begin{center}

\begin{tabular}{ |c||c| }

\hline

Name & Detail\\

\hline

Differential& Yes\\

Supported devices& 32 transmitters/32 receivers\\

Operation mode & Half-duplex\\

Longest supported distance & at 100kbps: 1200 meters\\

Highest supported transmitting speed& at 10Mbps: 16 meters\\

Mark (data = 1) condition& 1.5V to 5V (B greater than A)\\

Space (data = 0) condition& 1.5V to 5V (A greater than B)\\

Output current capacity& 250mA\\

Receiver input sensitivity& ±200 mV\\

Receiver input range& –7V to 12V\\

\hline

\end{tabular}

\caption{RS-485 Remarkable Specifications}

\label{table:RS-485HighLightSpecs}

\end{center}

\end{table}

%full-duplex, half-duplex implementation figure

\begin{figure}[!ht]

\begin{center}

\includegraphics[scale=0.75]{images/rs485-fullduplex-halfduplex.jpg}

\caption{Half-duplex(upper) and Full-duplex(below) implementations}

\label{fig:fullHalfDuplex}

\end{center}

\end{figure}

Working principle of RS-485 is different in comparison to other standard; Instead of using a zero ground as the voltage reference, which will cause noise over the communication length, it uses floating voltage between two wires of the signal pairs, A and B or (+) and (-). After transmitting, the receiver compares the different of voltage between two wires and achieved the correct data with the lowest noise may cause. Figure~\ref{fig:rs485Wave} illustrates an example of the RS-485 waveforms transmitting one byte which has Mark, Space, and Idle phases. In most network, there will be one node acting as the master and the rest work as the slaves. At this point, the master sends command frame over the connection, and all slaves receive the data, then each slave with different functionality will work as programmed with different received data and also response to the master as programmed. The best practical result is obtained with the use of twisted pair of wires because some of the noise current will flow in the opposite direction with the current in the cable. In case using the straight cable, the noise current flows straightly along the cable in the same direction which will cause a loop current. Combined with the twisted pairs of wires, the cable also comes with shield, which is an accepted approach to restrain the noise, is used in applications that need higher noise resistance.

As written in the introduction, RS-485 can connect with multiple transmitters and receivers in the same network. For instance, using input resistance around 12k \si{\ohm}, the numbers of devices can connect to the system is up to 32. Besides, with the connecters, this number can increase significantly to the number of thousands and the transmitting distance can be also extended to kilometers. In addition, the network implemented with RS-485 needs termination, usually with a 120\si{ohm} resistance at the end of two wires. This is applied to terminate or minimize the reflection in order to avoid the fraud of sending data. Furthermore, it usually included pull up and pull down resistors for fail-safe bias in each wire in case that any wire is not controlled by any device. When input voltage ranging from -200mV to +200mV, receiver understands as “undefined” state, which caused by several reasons such as system is shutdown, connection from receiver to network is lost, or cable has an open or short part. In this case, fail-safe biasing is applied in order to confirm that the receivers receive defined states only.

**MQTT**

MQTT is abbreviation of Message Queuing Telemetry Transport, which is a protocol laid in Application layer of OSI model. It is designed as a machine-to-machine and remarkably lightweight protocol that helps communication between constrained devices becomes effortless in comparison to other wireless protocols. In detail, its working principle based on publish and subscribe methodology in order to reduce the amount of transmitting data which leads to the reduction of used bandwidth, latency and power consumption.

A \gls{mqtt} system is the combination of a server, but usually named broker, and the clients, in which can acts as either a publisher, subscriber or both. One broker can have numerous clients connect to and each client can subscribe to any topic it is programmed. These subscribers are following and watching for the changes of data of the subscribed topics, once other clients which are defined as the publisher publishes message to the topics, then the broker distributed the payload of message to other clients who had subscribed to those topics. In this scenario, the publisher and subscribers do not need to know the information of each other, the only needed are the topics for publishing and subscribing.

Figure~\ref{fig:mqttEg} refers a simple example of a MQTT system consists of one broker and four clients, in which three clients are the publishers and one is the subscriber that subscribes to three topics the publishers publish to. To be specific, three publishers are the sensor nodes publish to three topics, temp1, temp2 and temp3; the last client named Sensor Data Gatherer acts as the subscriber that follows three topics mentioned previously. At once, the subscriber will receive the message whenever a publisher broadcasts the message to any of those topics. Implemented MQTT system will be described in details in later part of the thesis.

**WebSocket**

WebSocket was introduced in 2011 in term of a communication protocol, implementing full-duplex channels over one TCP connection. The protocol is laid in Application layer of OSI model but distinguished from HTTP. Distinct to HTTP, in WebSocket applications, a server can send the data to client without waiting for requests from client. All data are being sent between server and client will be sent to a settled connection which helps accelerate the data rate and keep the connection opened when necessary. Similar to \gls{mqtt}, it is designed to reduce the transmitting data, which leads to the reduction of bandwidth and consumed power. Although it is designed for web applications, still, it can be implemented in any applications that need such a lightweight protocol.

\begin{figure}[!ht]

\begin{center}

\includegraphics[scale=0.8]{images/ws-1.jpg}

\caption{WebSocket working principle}

\label{fig:wsPrinciple}

\end{center}

\end{figure}

Figure~\ref{fig:wsPrinciple} illustrates how WebSocket works. There are two parts of the protocol, handshaking and transmitting data. At first, client sends a request to server to initialize the websocket connection, the server then send an acceptance to connect. After this point, the data are being sent as WS frame with numbering as in the figure~\ref{fig:wsPrinciple}.