A Universal Microcontroller Circuit and Firmware Design and Implementation for IoT-based Real-time Measurement and Control Applications

Santi Nuratch
Embedded Computing and Control Laboratory
Department of Control System and Instrumentation Engineering, Faculty of Engineering
King Mongkut's University of Technology Thonburi Bangkok, Thailand
E-mail: santi.inc.kmutt@gmail.com, santi.nur@kmutt.ac.th

***Abstract*—This paper presents a design and implementation techniques to construct a universal microcontroller board based on real-time measurement and control applications. It is not only designed for real-time applications, but is also designed for supporting many modern WiFi devices that utilize TCP/IP protocol over internet networks. It means the proposed system fully supports many Internet-of-Things (IoT) applications. Furthermore, it can also be used for computer based monitoring and control applications and many other industrial applications, including Industrial Internet-of-Things (IIoTs) as well. In addition, the universal microcontroller board is designed for flexibility of use, meaning the users can custom the board to reduce and expand their applications. The system is composed of all necessary components including analog and digital input/output ports, WiFi communication based on TCP/IP. The real-time operating system (RTOS) and communication protocol drivers, e.g., UART, I2C, and SPI, are also embedded in the microcontroller firmware. The system is evaluated and examined in web-based real-time monitoring to prove that it can be used in IoT based real-time applications. The experimental results show that the proposed system appropriately exchanges data over internet networks to web-browser with high speed data transfer rate as expected.**

Keywords—microcontroller; circuit; firmware; internet-of-thing; measurement; control; real-time

I. Introduction

Nowadays we are living in the era of the "Internet-of-Things" (IoTs) and many companies are focusing and moving to a new technology of industrial automation called "Industrial 4.0" [4]. Combined both of them together, a new technology called "Industrial Internet-of-Things" (IIoTs), is defined and specifically used in the industrial automation systems. The IoT device is commonly designed and implemented to wirelessly exchange some desired information over internet networks. It means that an IoT device is basically composed of many modules, e.g., sensors, actuators, wireless communication device and microcontroller. With the electronic technologies today, we can buy many cheap electronic devices to build an IoT device. It is known that the brain of this kind of device is a microcontroller. Unfortunately, the microcontroller cannot work by just connect it to a power supply as other electronic devices. It requires a suitable program or firmware to execute its tasks. The firmware is commonly written using C-language

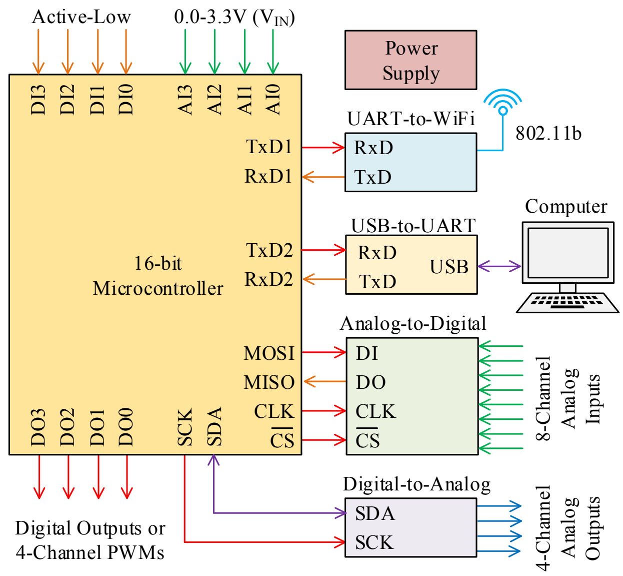

Fig. 1. Block diagram of the proposed system

and compiled by the specified compiler to generate a machine code. The machine code, which is sometimes called "hex code" or "instruction code", is then written to the program memory of the microcontroller. Furthermore other languages can be used to develop the firmware depended on programmer. Todays, the IoT devices are required in many applications [4]. Some simple applications can be constructed by ready-made devices, but generally these kind of ready-made devices are not designed for the complex tasks and they require higher cost.

To make a low-cost and plentiful functionalities that can be used in various applications, including research works, smart home, smart factories, real-time measurement and control systems, we design, implement and propose a universal microcontroller board for IoT applications. The board is specifically designed for the IoT based real-time measurement and control applications. In addition it is designed to support system flexibilities. Namely, it allows users to change input and output ports to fit to their applications. It also supports commonly used protocols used in embedded systems, e.g., UART, SPI and I2C. The users can custom, minimize or expand the system up to application requirements. The overview of the proposed system is descripted in Section II.

978-1-5090-4666-9/17/$31.00 ©2017 IEEE

The firmware implementation technique is illustrate in Section III. Section IV descripts experimental setup and results. Finally, conclusion is given in Section V.

II. The Proposed System

The proposed system is composed of many parts working together. The system is designed to operate with both analog and digital signals, including wire and wireless communication channels. A block diagram including data links between each module of the proposed system is clearly explained in Fig. 1.

A. External Peripheral Modules

Although the microcontroller used in this research has plentiful built-in peripherals [6], but sometimes it is not enough for real-world applications. For example, some applications require WiFi communication, serial port communication, many channels of ADCs and DACs. With these various requirements, we include the commonly used modules in the system as shown in Fig. 1. The details of the system are explained below:

Power Supply is a regulator circuit, the low voltage drop (LDO) regulators chip, the LM1117 [6]. It maintains output voltage at 3.3Vdc that required by the microcontroller and all peripherals in the system.

Microcontroller used in this work is a 16-bit MCU, PIC24FJ48GJ002 (Microchip Technology) [6]. It has plentiful peripherals required in complex microcontroller applications.

UART-to-WiFi is a TCP/IP supported WiFi module based on 802.11, the standard for wireless local area networks (WLANs) [1]. It has built-in microcontroller and wireless communication circuit that can significantly help developer to develop WiFi supported devices without any trouble. In this work, we select a low-cost module, the ESP8266 [5]. The module is programed (flashed) by a firmware supported by AI-thinker. With this firmware, the module can work as an AT command supported device [5]. After the module works as an AT command device, the microcontroller can fully control or request all parameters of the module by using a serial port (UART). It implies that the microcontroller can wirelessly exchange data over internet networks through its serial port.

USB-to-UART is a protocol translation circuit. In this case, it converts the UART protocol of the microcontroller to the USB protocol of the host computer. This module is commonly used in microcontroller's firmware development and run-time debugging because the microcontroller can send all information (variables) to a development computer making developer knows all desired parameters at run-time executions. Furthermore it is required for computer and microcontroller interfacing applications, e.g., computer-based monitoring and control applications. In this work, the FT232RL [2] is used.

Analog-to-Digital is a circuit that used to convert analog voltage to digital data. In this system, an 8-channel, 10-Bit ADC Converters with SPI Serial Interface, MCP3008, is used [6]. It is a successive approximation ADC converters with on-board sample and hold circuitry. The module can be programmed to provide four pseudo-differential input pairs or eight single-ended inputs. The microcontroller can communicate to this device using a simple SPI protocol (explained in next sub-section). The MCP3008 support high conversion rates up to 200 kilo-samples per second (ksps), fast enough for real-time measurement and control applications.

Digital-to-Analog helps the microcontroller to generate analog voltage for other devices. It is commonly known that the microcontroller can generates Pulse Width Modulation (PWM) signal and pass the PWM signal to an appropriate low-pass filter constructed by simple RC circuit. The filtered PWM signal, the output of the low-pass filter, is the analog voltage and the voltage level can be changed by adjusting the duty ratio the PWM. Theoretically, filtering the PWM signal by the low-pass filter circuit makes signal ripple. Increasing the order of the low-pass filter can reduce the ripple, but it increases phase delay. In additional, this method cannot be used to generate signals that have high dv/dt or fast changing waveforms, e.g., step-change and stair signals. Because of those reasons, the external DAC module, the MCP4728 [6], is used. The MCP4728 device is a quad, 12-bit voltage output DAC with nonvolatile memory (EEPROM). It has on-board precision output amplifier allowing it to achieve rail-to-rail analog output swing. The microcontroller can communicate to this device using I2C serial interface protocol, explained in next sub-section.

B. Communication Protocols

Considering in the system overview shown in Fig. 1, there are three standard communication protocols, Universal Asynchronous Receiver/Transmitter (UART), Serial Peripheral Interface (SPI), and Inter-Integrated Circuit (I2C). The first one is UART protocol used for WiFi module and computer interfacing. In this work, the parameters of the two UARTs are set as following: BaudRate=115200, StartBits=1, DataBits=8, StopBits=1, and no Parity and Handshake are used. The second one is SPI protocol used for the 8-channel ADC chip, the MCP3008. Other devices that supports SPI protocol communication can be connected to the same SPI bus by separating the CS line. The other lines, i.e., MOSI, MISO and CLK, can be shared to other devices. The last one is I2C protocol used for the 4-chammel DAC chip, the MCP4728. The I2C devices have their own programmable address. Like the SPI bus, the I2C bus allows multiple I2C devices (different addresses) connect to the same bus. The data exchanging of these protocols (UART, SPI, I2C) are develop by queue-driven, the ring-buffers to improve efficiency of the communication. The three protocols are supported by the built-in peripheral modules of the microcontroller, the PIC24FJ48GA002.

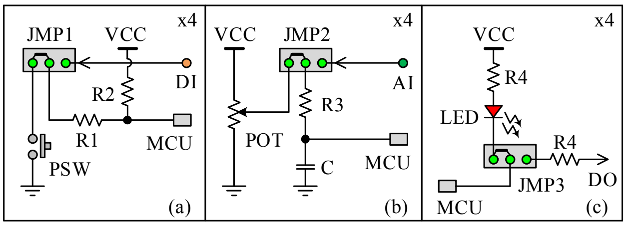

Fig. 2. On-board input and output circuits

C. On-board Input and Output Components

To construct the system that can be applied to various applications and can be used in both analog and digital signals, the input and output pins of the microcontroller are needed to be carefully designed to connected to external circuits appropriately. In this work, three basic types of components are included in the board. Firstly the digital output components (4 LEDs). Secondly, the digital input components (4 PSWs). Finally, the analog signal generators (4 POTs). Each component is connected to the input or output pin of the microcontroller through jumper, explained in Fig. 2. Note that the DI, AI and DO are connected to extended connectors. It means that the input/output pins of the microcontroller can be independently connect to either on-board components or external circuits. This design technique make the user become flexible and easy to construct the experimental system conveniently.

III. Firmware Implementation Technique

The main part of the system is the microcontroller that requires an appropriate firmware to run. In this research the firmware is implemented to work as a real-time operating system (RTOS). The RTOS is implemented based on non-preemptive scheduler, small footprint and fast response [2], [3]. In addition all drivers for built-in peripherals (ADC, OC, PWM, and others) and communication protocols (SPI, I2C, UART) are also embedded into the firmware. For the SPI and I2C communication operations, the data queues, the ring-buffers, are implemented for both transmitting and receiving operations. The queues are efficiently and fully controlled and maintained by the RTOS. For the serial port communication for both WiFi module (URRT1) and computer (UART2), the interrupt even-driven and queues are implemented. Each UART module (serial port) has its own interrupt events, e.g., Data Received (RxD) interrupt event and Data Transmitted (TxD) interrupt event. Using these events together with queues or ring-buffers can significantly improve real-time communication abilities of the system, also this technique guarantees that all bytes of data will be transmitted and received on time and confirms that no data loss [2].

Another important module that needed to be implemented and embedded into the firmware is TCP/IP protocol for real-time data exchanging [1]. In this work, the WebSocket is selected because it is designed for real-time communication over internet networks. It is also the client-server topology communication. The WebSocket supports full-duplex communication over TCP/IP. Meaning that either side of the connection can send data to the other, even at the same time [1]. In this research the microcontroller is implemented to work as WebSocket server and exchange data with client (web browser) through WiFi module.

IV. Experimental Setup and Results

To evaluate and examine all parts of the proposed system, we firstly design a simple circuit as shown in Fig. 3. This circuit is designed to evaluate the I2C (DACs) and SPI (ADCs) devices. Each output voltage of the DAC device is connected to an RC low-pass filter and each output of the low-pass filter is connected to input of the ADC device. The resistance of the resistors R_1 to R_4 are 20kΩ, 47kΩ, 72kΩ and 100kΩ respectively. The capacitance of all capacitors, C_1 to C_4, is 1.0µF. The microcontroller controls the DAC device to generate a step-changed signals on the output pins (DAC_0 to DAC_3). In this experiment, the signals appearing on the DAC_0 to DAC_3 have the same waveform as shown in Fig. 4. (black line, $V_{in(t)}$). Note that the black line is measured directly on DAC_0 pin, connected to ADC_0 of the SPI device. The microcontroller also reads the voltage across C_1 to C_4 (the outputs of the low-pass filters) by sending request command to the ADC device. The signals appearing on the C_1 to C_4 (ADC_1 to ADC_4, $V_{C1(t)}$ to $V_{C4(t)}$) are shown in Fig. 4, represented by the red, green, blue, and pink lines respectively. The sampling rate of this experiment is 500 Hz. The waveform of the step-charged is changed every 100 mS ($5R_1C_1$).

Another circuit is constructed to evaluate the built-in PWMs (OCs) and ADCs modules of the microcontroller. The circuit for this experiment is shown in Fig. 5. All resistors and capacitors are the same as the previous experimental circuit, but the capacitor C_1 is removed. In this experiment, the microcontroller generates the same waveforms of PWM signals on the PWM_0 to PWM_3. These signals, except PWM_0, are connected to the RC low-pass filters. The output of the low-pass filers (voltage across C_2 to C_4) are used as input signal of the ADC_1 to ADC_3. Note that there is no filtering between the PWM_0 and ADC_0. To evaluate the system, the microcontroller generates PWM signals that have 40 Hz of frequency. The first 20 cycles of the PWM signals (PWM_0 to PWM_3) have 70% duty cycle, then they are changed to 30% duty cycle.

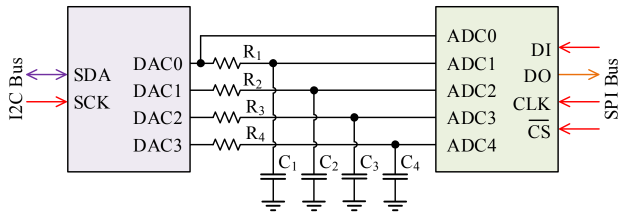

Fig. 3. Circuit used to evaluate the I2C (DACs) and SPI (ADCc) devices

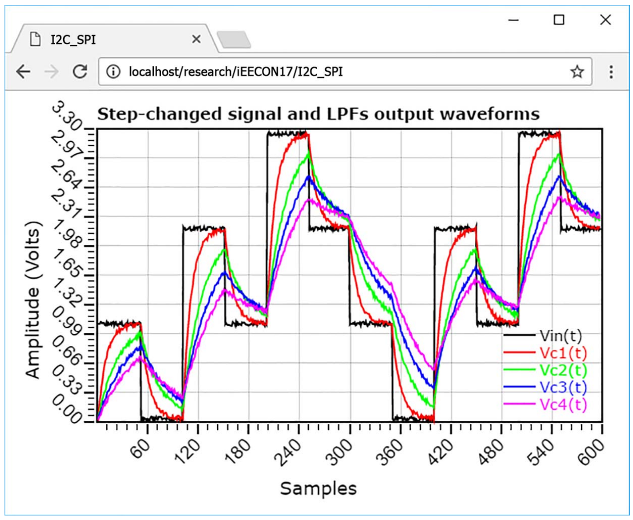

Fig. 4. Waveforms of the step-changed and outputs of the low-pass filters

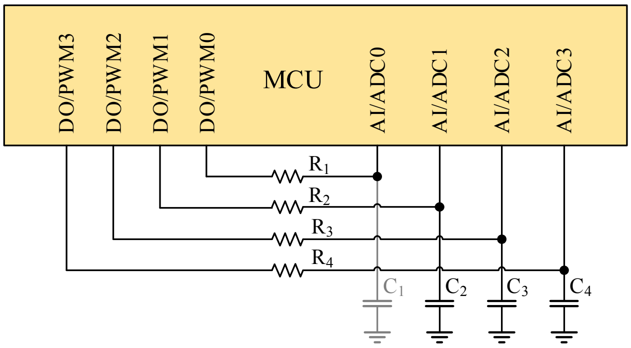

Fig. 5. Circuit used to evaluate the built-in PWM and ADC modules

All signals are sampled at 1.0 kH of sampling frequency. The waveforms of the measured points are shown in Fig. 6. The gay line is the waveform of the PWM signals representing the signals of the PWM_0 to PWM_3 (they have the same shape). The output waveforms of the low-pass filters, the voltage across C_2 to C_4 are visualized as the red, green and blue lines respectively.

Note that the web-based applications used to visualize the waveforms are implemented with HTM5 and JavaScript [4]. The waveforms displayed on the web-browser (shown in the Fig. 4 and Fig. 6) are updates in real-time like the display of oscilloscope.

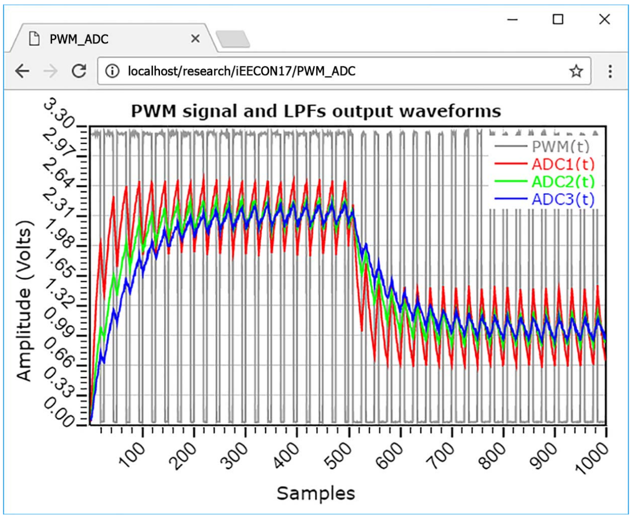

Fig. 6. Waveforms of the PWMs and ADCs (outputs of the low-pass filters)

V. Conclusion

The design and implementation techniques to construct the universal microcontroller board, including circuits, protocols, and experimental setups and results are reported in this research. Both of the proposed hardware and firmware are mainly designed and implemented for real-time measurement and control applications and the IoT applications. The experimental results show that the proposed system can exchange data over the internet networks in real-time data streaming as expected. Not only the conceptual design techniques are proposed, but we developed the universal microcontroller circuit (board) for researchers, engineers and students. The board is shown the Fig. 7. In the future, we will analyze and report the system in-depth details, e.g., event-driven technique, real-time communication over internet networks like WebSocket and MQTT, and construct a cyber-physical system (CPS) for web-based real-time measurement and control to support the Industrie 4.0.

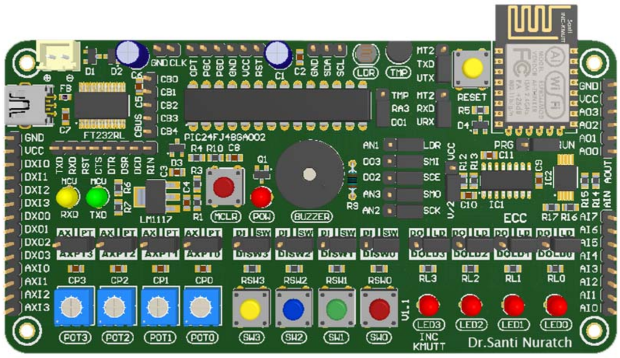

Fig. 7. Circuit used to evaluate the built-in PWM and ADC modules

References

[1] Fette, Melnikov, "Relationship to TCP and HTTP," RFC 6455 The WebSocket Protocol. IETF. sec. 1.7. RFC 6455, December 2011

[2] M. Barr and A. Massa, "Programming Embedded Systems", Second Edition, O'Reilly Media, Inc., 2006

[3] Phillip A. Laplante Seppo J. Ovaska, "Real-Time Systems Design and Analysis: Tools for the Practitioner", ISBN: 978-0470768648, 2012

[4] Tan Lu, Wang Neng, "Future Internet: The Internet of Things," International Conference on Advanced Computer Theory and Engineering (ICACTE) vol. 5, pp. 376–380, August 2010.

[5] https://espressif.com

[6] http://www.microchip.com/