Master Thesis

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Measurement

Learning and automation GPIO platform

Ondřej Hruška

Supervisor: doc. Ing. Radislav Šmíd, Ph.D.
Field of study: Cybernetics and Robotics
Subfield: Sensors and Instrumentation
2018

MASTER'S THESIS ASSIGNMENT

CTU CZECH TECHNICAL UNIVERSITY IN PRAGUE

I. Personal and study details

Student's name: **Hruška Ondřej** Personal ID number: **420010**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Measurement**

Study program: **Cybernetics and Robotics**

Branch of study: **Sensors and Instrumentation**

II. Master's thesis details

Master's thesis title in English:

Learning and Automation GPIO Platform

Master's thesis title in Czech:

Výuková a automatizační GPIO platforma

Guidelines:

Design and implement a modular system consisting of a motherboard and additional modules for connecting sensors, actuators and general inputs via I2C, SPI, UART, 1-Wire or other interfaces to the central system via USB, UART, and wireless interfaces. Allow access to built-in processor peripherals such as ADC, DAC, and timers (PWM, frequency measurement). Design a comfortable way to set the configuration without firmware changes. For the designed system, create a service library in C, Python, and MATLAB.

Bibliography / sources:

[1] STMicroelectronics datasheets, http://www.st.com

[2] Ganssle, J.: The Art of Designing Embedded Systems, Elsevier Science, 2008.

[3] Chi, Qingping & Yan, Hairong & Zhang, Chuan & Pang, Zhibo & Da Xu, Li. (2014).: A Reconfigurable Smart Sensor Interface for Industrial WSN in IoT Environment. Industrial Informatics, IEEE Transactions on. 10. 1417-1425. 10.1109/TII.2014.2306798.

Name and workplace of master's thesis supervisor:

doc. Ing. Radislav Šmíd, Ph.D., Department of Measurement, FEL

Name and workplace of second master's thesis supervisor or consultant:

Date of master's thesis assignment: **10.01.2018** Deadline for master's thesis submission: ________

Assignment valid until:
by the end of summer semester 2018/2019

doc. Ing. Radislav Šmíd, Ph.D. Supervisor's signature	Head of department's signature	prof. Ing. Pavel Ripka, CSc. Dean's signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

Date of assignment receipt — Student's signature

Declaration

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 27. května 2018

....

Acknowledgements

blabla

Abstract

This work focuses on the design of an AC appliance degradation detector. The goal is to implement a device in the form of a power-plug adapter that could be used to monitor and study the characteristics of the AC current.

A prototype with a STM32 F3 processor and an ESP8266 programmable WiFi module has been realised, together with a custom firmware for both processors, which allows easy access to the measurements and charts using a web browser. The device also supports regular reporting to a Xively or ThingSpeak monitoring server.

Keywords:

Supervisor: doc. Ing. Radislav Šmíd, Ph.D.

Abstrakt

Tato práce se zabývá implementací detektoru poruch a degradací síťového spotřebiče pomocí analýzy časového průběhu odebíraného proudu. Cílem je navrhnout a realizovat přístroj ve formě zásuvkového adaptéru, který by bylo možné použít k monitorování připojeného zařízení.

V rámci práce byl realizován prototyp přístroje s procesorem řady STM32 F3 a programovatelným WiFi modulem ESP8266. Zařízení umožňuje pohodlné ovládání a zobrazení grafů spektra a průběhu proudu pomocí webového rozhraní. Dále je podporováno pravidelné hlášení stavu na servery Xively a ThingSpeak.

Klíčová slova:

Překlad názvu: Výuková a automatizační GPIO platforma

Contents

Figures

Chapter 1

Introduction

Prototyping, design evaluation and the measurement of physical properties in experiments make a daily occurrence in the engineering praxis. This task involves the generation and capture of electrical signals coming to and from specialized sensors, actuators and other circuitry. As the technology advanced, driven by the consumer electronics market in the pursuit of miniaturization and lower cost, a multitude of integrated sensors with digital interfaces became available. Those devices can provide a sufficient accuracy and precision for the task at hand while keeping the circuit complexity down by integrating a large portion of the necessary circuit, often including the sensor itself, on a single chip or in a compact module.

The drive for miniaturization and the advent of modern hardware buses, in particular USB (Universal Serial Bus), however, simultaneously lead to the disappearance of low level computer ports, such as the once ubiquitous parallel port, that would provide an easy way of connecting those digital devices and low level hardware in general to personal computers.

With those developments, we on the one hand have an easy access to interesting integrated devices that would make the mentioned experiments and evaluations easier, on the other, our means of interfacing them have become more complicated. When an engineer today wants to perform some measurements using a digital sensor, they are often left having to implement an embedded firmware for a microcontroller that can be connected to the PC through USB, commonly through a USB-USART adapter. This approach lets one optimize the solution for a particular task and possibly achieve higher performance, however it is time-consuming and requires knowledge of the embedded system design that is entirely removed from the desired measurement.

Clearly it would be advantageous to have a way to easily attach those integrated devices and low level hardware in general to a PC without having to burden ourselves with technicalities of the connection. The design and implementation of such a system is the object of this work.

Chapter 2

Goals and Requirements

This chapter analyzes the project requirements and presents a vision of the final outcome.

2.1 Project Name

Referring to the project only as "the project" or "the device" would be clumsy and lead to confusion. Henceforth, the project should be known as **GEX**, an acronym originating in the term *GPIO Expander*, which, although not describing its scope perfectly, alludes to the project's primary purpose of providing low level GPIO capabilities to personal computers.

2.2 Expected Use-Cases

A first step must necessarily be to consider the situations in which the device is expected to be useful. As was mentioned in the introduction, one of the problems addressed is making it possible to easily attach electronic circuits and components to a PC. This could be in order to familiarize oneself with a new chip or a module, to measure a characteristic curve of a component, to collect experimental data from a test setup, or, for instance, to control a positioning motor.

The applications can have a temporary character, a simple setup that is used once and then dismantled, or a more permanent one. An example of the latter could be laboratory tasks where the measurement framework and user interface is prepared beforehand and connecting the circuit or simply performing measurements while varying some physical properties is left as an exercise to students. Another example could be the use of GEX as a data acquisition module, replacing more expensive professional devices.

As such, the device must be easy to configure without having to modify the embedded firmware and should provide the hardware interfaces and functions that are be needed for such applications. The module can be either attached directly to a PC via USB or controlled wirelessly. It would also be possible to design it as a PCI Express card, however that would limit its use to desktop computers and make the installation and software support more complicated. The wireless connection will find use in mobile robotic projects, when installed in less accessible places, or outdoors.

2.3 Hardware Interfaces to Implement

Given the project's broad range of potential applications, predicting precisely what hardware interfaces and connections might be needed is hardly possible. A good approach appears to be to implement the most common protocols and interfaces and provide access to selected low level features offered by the used microcontroller, like timers and direct pin access, while keeping the firmware open to future expansions should the need arise.

2.3.1 Direct Digital Input/Output

The most basic form of interaction with hardware is by changing the logic levels of output pins and reading input pins. With this feature alone it would be possible to analyze logic circuits, trigger some transient effect we want to observe using an oscilloscope, read a contact state, sense a button push, drive LED displays and more. Almost anything digital that doesn't require precise or fast timing could be achieved by this simple function.

To make this feature more versatile, it should be possible to receive an asynchronous event on a pin state change, avoiding the need for polling loops in the control application.

2.3.2 Common Digital Buses

A popular way to attach peripheral devices to a microcontroller are hardware buses, the most well known of which are SPI (*Serial Peripheral Interface*), I^{2}C (or IIC, *Inter-Integrated-Circuit*) and USART (*Universal Synchronous Asynchronous Receiver Transmitter*), in the asynchronous variant referred to as UART. A large majority of peripheral integrated circuits, digital-interface sensors and modules can be accessed using those buses. They also have hardware support in most microcontrollers, removing the burden of precise timing from the firmware.

Another hardware interface that might fall into this category is I^{2}S (or IIS, *Inter IC Sound*). I^{2}S is used for streaming sampled audio streams or analog samples and finds far less use than the previously mentioned buses.

link to actual place

More information about those interfaces (excluding I^{2}S) can be found in later chapters.

2.3.3 Specialized Buses

source

Some devices exist that do not use any of the common buses, instead requiring their own proprietary protocol. An example of this group is the Dallas Semiconductor[1] 1-Wire bus, used by the popular DS18x20 digital thermometers. Another example might be some types of addressable LED strips.

A common characteristic of those buses is that they require precise timing and have no native hardware support like the above mentioned common buses. This means that we can't depend on the direct GPIO access and have to implement low level driver utilities to achieve reliable communication.

[1] Acquired by Maxim Integrated in 2001

2.3.4 Analog Input/Output

Microcontrollers typically include an ADC (*Analog to Digital Converter*) with a 10-12 bits of resolution, sometimes accompanied by a DAC (*Digital to Analog Converter*), its output counterpart. In the lack of a real DAC, the analog output, albeit with worse dynamic parameters, can be realized using a PWM signal (*Pulse Width Modulation*, pulse train) followed by a low-pass filter.

While we mainly focused on digital interfaces thus far, providing means of generating and capturing analog signals is also valuable. This capability makes it possible to read sensors with voltage output and it can substitute a simple oscilloscope when sampled periodically at a sufficient frequency. Analog input channels, even with lower resolution or sample rate, may in some cases avoid the need for a dedicated acquisition device.

In conjunction, the analog output and input can be used for automated characterization of electronic components, such as diodes. Should the analog output be modulated, we could further use them to measure frequency-dependent characteristics, such as the frequency response of analog filters.

Chapter 3

Existing Solutions

The idea of making it easier to interact with low level hardware from a PC is not new. Several solutions to this problem have been developed over the past years, each with its own advantages and drawbacks. Some of the existing solutions will be presented in this section.

3.1 Bus Pirate

pictures

http://dangerousprototypes.com/blog/about/

Bus Pirate, developed by Ian Lesnet at Dangerous Prototypes and manufactured by Seeed Studio, is a "tinkering kit" providing access to hardware interfaces like SPI, I^{2}C, USART and 1-Wire (those will be described later), as well as frequency measurement and direct pin access.

link

link

link to actual place

The board aims to make it easy for the user to familiarize themself with new chips and modules; it also provides a range of programming interfaces for flashing microcontroller firmwares and memories. It communicates with the PC using a FTDI USB-serial bridge

Bus Pirate is open source and in scope it is similar to what we want to achieve here. It can be scripted and controlled from languages like Python or Perl, connects to USB and provides a wide selection of hardware interfaces.

The board is based on a PIC16 microcontroller running at 32 MHz. Its analog/digital converter (ADC) only has a resolution of 10 bits (1024 levels). There is no digital/analog converter (DAC) available on the chip, making applications that require a varied output voltage more difficult. Another limitation of the board is its low number of GPIO pins which may be insufficient for certain applications, such as multi-channel sampling, parallel interfaces, or connecting multiple different devices at once.

Those limitations, however, hardly impede on Bus Pirate's primary purpose, which is to provide an easy access to digital buses.

3.2 Raspberry Pi

link, pictures

Another device worth mentioning, albeit of a very different kind, is the Raspberry Pi. It is a belief of the author that the inclusion of a GPIO (*general purpose I/O*) header on the Raspberry Pi mini-computers was a significant factor in their success in the hobbyist circles and school environments. This GPIO header exposes various hardware interfaces to user programs running on the computer.

The responsibility of controlling the experimental hardware then lies on the user application which also provides the user interface, much simplifying the development process. The control application can be written in almost any programming language the experimenter chooses; the most popular choices appear to be Python and JavaScript. The embedded firmware, should an external microcontroller be used instead, would typically have to be written in C, C++, or assembly.

A disadvantage of using a Raspberry Pi's GPIO header is that the experiments would have to be conducted directly on the mini-computer instead of using the more powerful computer the researchers already have available[1]. This introduces complications with data export or remote control. Further, should the experiment use a software package like MATLAB, installing it on the ARM-based Raspberry Pi may prove problematic.

3.3 The Firmata protocol

links

Move this elsewhere

Firmata is a serial communication protocol based on MIDI (*Musical Instrument Digital Interface*) for passing data to and from embedded microcontrollers. MIDI is primarily used for attaching electronic musical instruments, such as synthesizers, keyboards, mixers etc., to each other or to a PC.

Firmata was designed for use with the Arduino firmware to allow easy construction of user programs (called *sketches* in the Arduino environment) that communicate with a client application running on the PC without having to worry about technical details.

citation

Implementing the Firmata protocol in a universal hardware interfacing module would make it possible to use existing Firmata client libraries. However, it is constricted by the limitations of the encompassing MIDI protocol and offers little flexibility.

3.4 Professional DAQ modules

There are several offerings from professional laboratory instrument manufacturers, however their common property is a very high price. This renders them inaccessible for users with a limited budget, such as hobbyists or students who would like to keep such a device for

[1] An exception may be the use of such a device in developing countries, where the Raspberry Pi serves as a low-cost PC on its own.

personal use. An example falling into this category is the National Instruments "I²C/SPI Interface Device", which also includes several GPIO lines.

http://www.ni.com/en-gb/shop/select/i2c-spi-interface-device

The decoding of hardware buses like USART, SPI or I²C is a common feature in digital storage oscilloscopes, as is the sampling of digital channels with "logic analyzer" add-ons. They are valuable debugging tools, but hardly ever provide a way to interact with the bus beyond passively intercepting an ongoing communication.