Galway - Mayo
Institute of Technology

Final year project

Easysleep

Zdenek Krousky

supervised by
Paul Lennon

April 11, 2020

Declaration

This project is presented in partial fulfilment of the requirements for the Degree of Bachelor of Engineering (Hons.) in Software and Electronic Engineering at Galway-Mayo Institute of Technology. This project is my own work, except where otherwise accredited. Where the work of others has been used or incorporated during this project, this is acknowledged and referenced.

Acknowledgement

I would like to extend my thanks to my supervisor Paul Lennon who made sure I stay on track with my project as well as to Niall O'Keeffe for his support in embedded part of the project. I would also like to thank my wife Caroline for her ongoing support through my studies.

Table of Content

1 Project background and motivation

Aim of the project, Why?

The goal of this project was to create a device that would help in resolving nocturnal enurism (bedwetting) common in children above the age of 5. I am myself a parent of a child with such difficulty and can relay to the stress this causes. Me and my wife have tried various methods of resolving this such as encouriging my daughter going to the toilet prior to going to the bed, waking her up during the night and medication.

According to my research there is a number of different causes for this bedwetting to happen. Medical reasons only account for close to 3%. 15% of children above the age of 5 still wet the bed at night and up to 5% above the age of 10 continue to do so. The selection of night-time pants in grocery stores show sizes for children up to the age of 14.

I have come to the conclusion, that the main cause of my child being wet every night is the deep sleep. Her urge to go to the toilet simply isn't strong enough to wake her up at night.

This is where my project comes in. It aims to resolve the nocturnal enurism of children where the deep sleep is the root cause in somewhat gentler and smarter way than a bed-wetting alarm.

2 Overview

What is Easysleep?, Research, Architecture diagram

The project consists of two devices that are able to communicate via Bluetooth and a mobile phone application. These are 2 FRDM-K64F microcontrollers. One assumes the role of a master and monitors the moisture detection sensor, records the time and date of the event, sends notification to the secondary device to wake the sleeping person up and if requested, transfers time and date of last 10 events to the mobile phone application. It also notifies a parents or a guardians phone if an event happens and activates the buzzer on the sleeping persons bracelet.

The secondary device (slave) uses a Parallax vibration motor to wake the sleeping person up. If this is not acknowledged by pushing a button (indicating that he/she is awake) a second stage is entered and a buzzer is sounded.

The mobile phone application allows the user to silence an ongoing alarm remotely, request event data from the master device and save those into an SQLite database and check/change the system time and date.

3 Hardware

Hardware used, Connections, Specifications

My project doesn't consist of too many parts. Aside from the actual microcontroller it only requires a vibration motor, relay and buzzer all of which are used to wake the sleeping person up. HC-05 bluetooth modules are used to facilitate communication between the master device and the bracelet a mobile phone.

3.1 FRDM-K64F

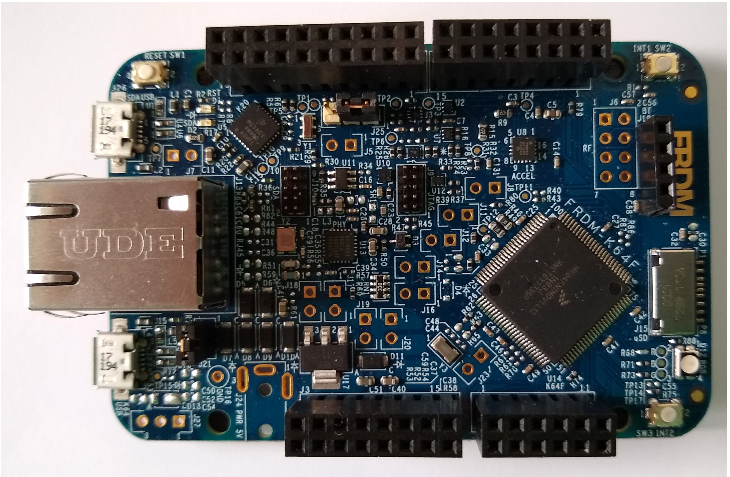

Figure 1: FRDM-K64F development board

FRDM-K64F is a very capable development board manufactured by NXP Semiconductors with a headquarters in Eindhoven, Netherlands and Austin, Texas. I have chosen this board because of my familiarity with it and of its abilities. This board and its cousin KL25Z were used throughout the course as part of the embedded systems modules.

Board specifications:

- 120MHz ARM Cortex-M4 microcontroller
- 1MB Flash memory
- 256kB RAM
- Ethernet

- SDHC
- low-power
- FXOS8700CQ accelerometer and magnetometer
- Add-on Bluetooth module: JY-MCU BT board V1.05
- RGB LED
- 2x user push buttons
- form-factor compatible with Arduino Uno Rev.3 pin layout

3.2 Parallax 28821 Vibration motor

The vibration motor is used to gently and quietly wake the sleeping person up in the first stage of the overall wake-up process. Parallax vibration motor seemed appropriate device for this task as it requires only 3V of power. However, the current it requires is quite high and cannot be supplied by the K64F thus an external supply has to be used.

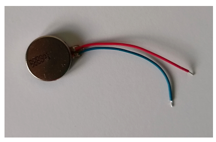

Figure 2: Parallax vibration motor

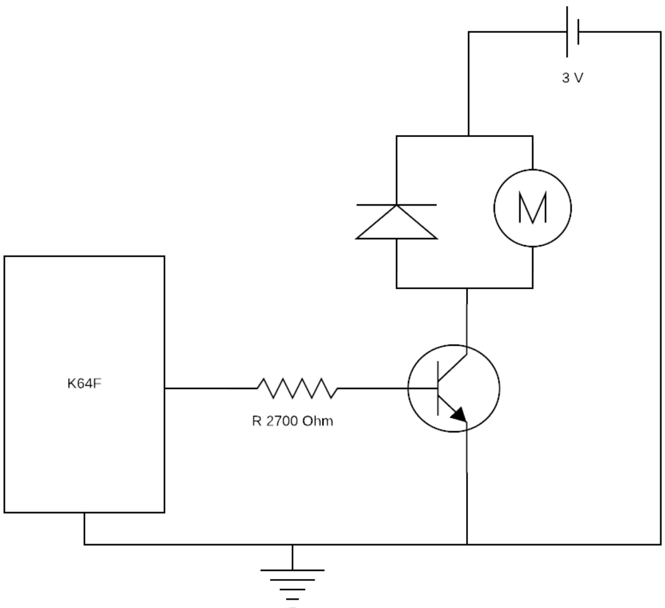

Figure 3: Vibration motor connection diagram

Motor specifications:

- Rate voltage: 3.0V
- Rate current: 150mA
- Rate speed: 9,000r/min Min
- Starting voltage: 2.3V

3.3 Buzzer

The buzzer is used for the second stage of wake-up process if the vibration motor or relay fail to wake the sleeping person. This is a passive device and for the sound to be produced the Flex Timer Module of K64F is used to generate PWM signal on the connected GPIO pin.

3.4 HC-05 Bluetooth module

Figure 4: HC-05

HC-05 is commonly used Bluetooth module often used with Arduino projects. The K64F development board supports this module also. I have installed small section of header pins to house this module. For my project I am using two of these. One to communicate with the secondary device (bracelet) and second one to communicate with the mobile phone.

This module can operate as a master or as a slave. Unfortunately it only supports one-to-one communication and that was the reason for adding the second module.

3.4.1 Bluetooth module configuration

In order for the Bluetooth modules to communicate both, the master and the slave have to be configured in the same way. Additionally they also have to share the same password and have to be made aware of each other by whitelisting each others address through AT commands.

To setup this configuration I had to use an FTDI cable and start the individual modules in AT mode. This is done by applying power to KEY pin prior to the VCC pin. To communicate with the Bluetooth modules and configure it in the way I needed, I have used **Minicom** - a Linux command line utility. I have started this by issuing the following command upon starting the bluetooth module in AT mode.

minicom -b 38400 /dev/ttyUSB0

The following commands were used to setup the master device:

- AT+UART=38400,0,0 (Baudrate, 1 stop-bit, no parity, default 8-bit mode)
- AT+ROLE=1 (master)
- AT+INQM=1,1,48
- AT+PSWD=7777

Similar setup was used for the second device with the exception of the ROLE, where it needs to act as a slave device therefore the ROLE had to be defined as 0.

4 Software

Software used, Programming languages, IDEs, Software tools
Througout my project I have used various software tools to develop C and Java code, as well as software to monitor the behaviour of the system.

4.1 MCUXpresso

I have used MCUXpresso 11.1 to develop code for K64F. This is an Eclipse based IDE tailored to suit NXP devices. There are lots of alternatives for other manufacturers such as Atollic Studio for STM32. I have became familiar with this IDE through my time in the college as it was used for code development in embedded systems classes. I have also used this IDE during my work placement in Jaguar LandRover.

The project uses Amazon FreeRTOS to efficiently manage various tasks of the system. This is an open-source real-time operating system and it's usage greatly reduces the complexity of C code used for functionality of the system. A programmer can divide the code into smaller, easier to manage blocks known as tasks. A communication between those is facilitated by usage of semaphores, notifications and queues.

Prior to commencement of work on the project I had to obtain appropriate Software Development Kit. I did so, through SDK Builder present on NXP website. This site allowed me to select specific processor and middleware and generated an SDK package that I then imported into the MCUXpresso. This package included all necessary drivers for various peripherals present on the development board such as GPIO or FlexTimer driver.

I also had to create a repository on Github to have a proven record of my work required by my supervisor and other lecturers and also to have a safety net in case things go wrong. This is an amazing tool to know and use. I was grateful to have the ability of reverting to previous revisions of my code throughout the development as few times the path I have chosen to steer the development proved unsuitable and it would have been impossible to revert changes from memory.

4.2 Android Studio

This integrated development environment came as the most reasonable choice for development of my mobile application. There are other options out there such as Eclipse IDE but the Android Studio is the most supported one and was actually used during my last semester in Mobile application development module so I was already familiar with it.

This IDE has been developed by BrainJet, studio behind well known IntelliJ and one can really see the similarities of both environments. This is an advanced IDE with plenty of features fully supported by Google Inc.

The idea behind this application was to allow the user to wirelessly control the Easysleep module. This communication allows the user to check Easysleep's time and date or configure those as well as request data of recent incidents and silence an ongoing alarm.

During my project development I have used a third party application that allowed me to send characters via Bluetooth to Easysleep and receive data back. I wanted to recreate this application in a way that would better suit the project.

5 Code Development

Languages used, various routes, code development and examples

The programming languages that were used for the development for the code were the C programming language and Java. C was used to program the two K64F embedded devices while Java was used for the Android application development.

5.1 K64F Master code development

K64F Master Device code repository

5.2 K64F Slave code development

K64F Slave (Bracelet) Device code repository

5.3 Android application code development

Android code repository

5.3.1 User Interface Development

GUI

5.3.2 Bluetooth Communication

BT

5.3.3 Database Development

???

The development of this application has been challenging and I had to follow a number of tutorials online in order to establish the Bluetooth communication. I have decided that for the purpose of this document I will divide the development process described into three subsections:

- User Interface Development
- Bluetooth Communication

- Database Development

5.4 Other

Additional software tools used

5.4.1 SystemView

SystemView is a great tool developed by SEGGER for monitoring the behaviour of Real-time operating systems as well as interrupts. It is free to use. It requires J-Link to be either installed as a bootloader on the development board or a harware J-Link probe needs to be used. In my case I had an access to the hardware version. In order for it to work fully, I had to cut two traces on the development board otherwise it would clash with the OpenSDA debugger and the code would not get uploaded to the memory.

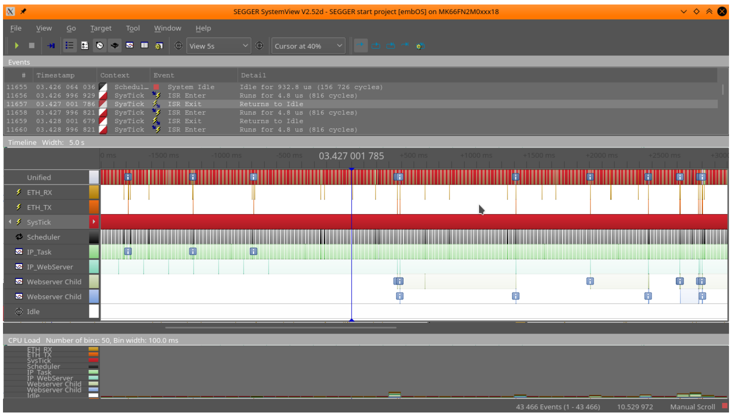

Figure 5: SystemView

5.4.2 FreeRTOS

FreeRTOS is one of many real-time operating systems available. Other options include QNX, ThreadX, embOS or Zephyr. Using a real-time operating system allows the programmer to divide the functionality of the project into individual blocks that are easier to manage, can communicate amongst themselves and maintain responsiveness. Real-time operating systems are often used in places where functionality of a system is somewhat complex and certain level of responsiveness is required (time-deterministic).

5.4.3 Pulseview

Pulseview is a Qt based logic analyzer. It is licensed under GNU GPLv.3. This software allows the user to monitor the output of GPIO pins as well as decoding of various protocols, such as SPI or UART. This package is in the repositories of my Linux distribution so installing it was as simple as typing a **sudo apt install pulseview** into the command line and let the system to do the rest. This tool has proven invaluable when I was working with buzzer and FlexTimer Module as it allowed me to monitor the output square wave.

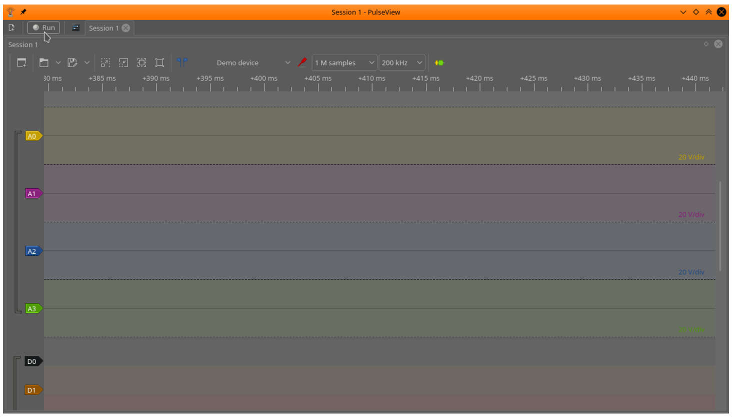

Figure 6: Pulseview

5.4.4 Git/GitHub, gitk

Throughout the development of the C code for the embedded platform as well as of the Android application and this report I have used a Version Control System known as Git and it's online counterpart GitHub. This allowed me to safely store my code, revert to previous code revisions and work from different computers at school or at home.

5.4.5 Doxygen

Doxygen is a documentation generator widely used in software development companies. It allows user to generate documents describing functionality of the code just by adhering to certain code commenting standards.

Doxygen is available for various programming languages. It comes with a large configuration file that the user can adjust to suit the need of the project.

5.4.6 Project management software

project management software, BT configuration,

6 Conclusion

what was the development of the project like

7 References

8 Bibliography

List of Figures