JevonAlarmClock : development report

Jeroen Vennegoor op Nijhuis

June 13, 2015

Abstract

This report provides indepth information on the JevonAlarmClock. JevonAlarmClock is an electronics project by Jeroen Vennegoor op Nijhuis and was developed for the 2015 Keil/Arm "ARM Microcontroller Design Contest".

Contents

Chapter 1

Introduction

In April 2015 I got an Elektor newsletter in my inbox. It mentioned the ”ARM Microcontroller Design Contest” organized by Keil/ARM. This sounded like the perfect opportunity to test and improve my technical skills in the field of electronics and embedded software. This document describes the development of the JevonAlarmClock: an Internet connected smart alarmclock that can control a Philips Hue light in order to wake you up in the morning.

Chapter 2

Architecture

2.1 System description

JevonAlarmClock consists of 2 parts: the alarmclock itself and a gateway that connects the alarmclock to the Internet. The alarmclock wirelessly communicates with the gateway via Zigbee. The gateway itself is via Ethernet connected to the Internet and to the local Philips Hue gateway. The alarmclock can actually control 1 Philips Hue lamp. This is achieved by send a command to the gateway where it is translated to a HTTP REST call to the Philips Hue gateway. The basic functions of the gateway are:

- provide current time to the alarmclock
- translate light control commands from the alarmclock to HTTP REST calls to a local Philips Hue gateway.

Chapter 3

Electronics

3.1 Introduction

The electronics for JevonAlarmClock has the STM32F429I-DISCO development at it's core. Several components needed to be added to provide the following functionalities

- Power supply
- RF communication
- Audio output
- User input(buttons)

3.2 Power supply

The JevonAlarmClock uses a standard wallsocket power supply to provide a 5V DC input. To reduce this to the 3V required for the XBee, an LM317T is used to convert 5V DC to 3V DC. The LM317T is an adjustable voltage regulator: the output voltage can be configured by choosing the correct resistor values. The combination of a 470 Ω resistor and a 660 Ω gives an output of 3V.

3.3 RF communication

For wireless communication between the JevonAlarmClock and the gateway, an RF module had to be added. In this case an XBee Zigbee module was chosen. The communication between the ARM microcontroller and the XBee is via UART (3V level) using only RX and TX at 9600 Baud.

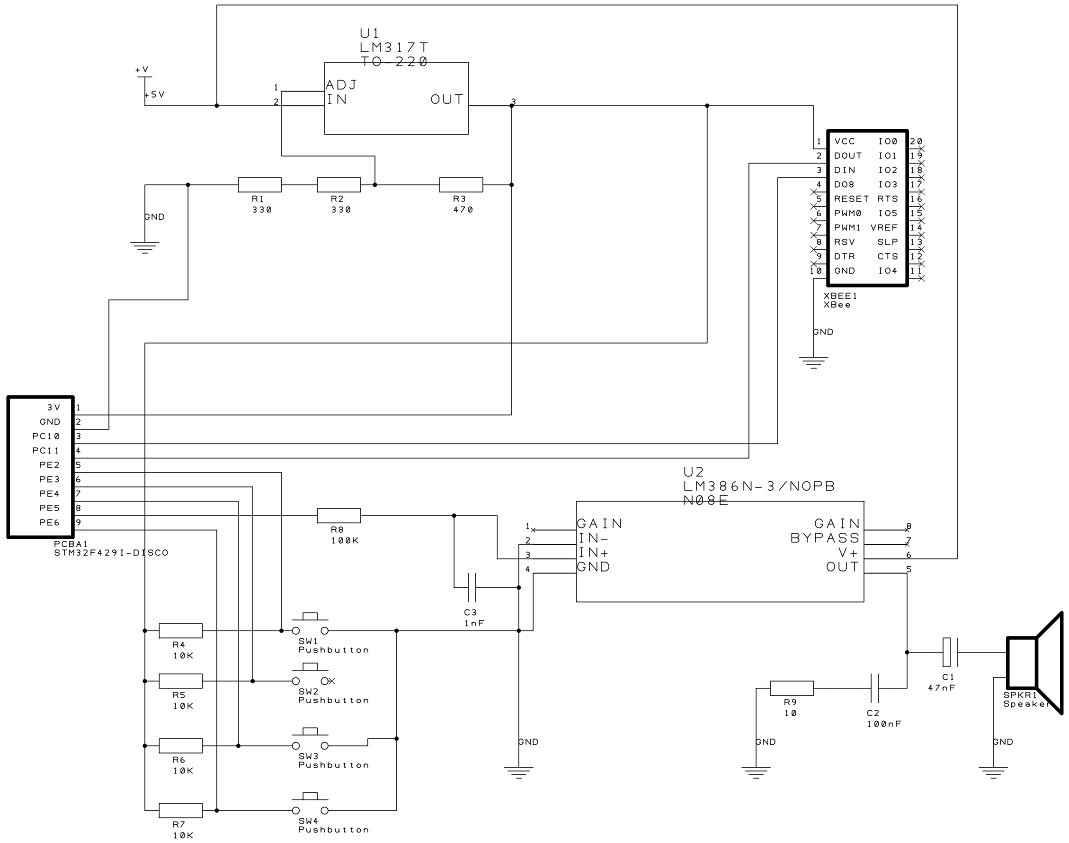
3.4 Audio output

The STM32F429 doesn't provide any audio output functionality, instead 1 GPIO pin was configured as PWM output and a basic audio amplifier was added to create sufficient power for a small 8 Ω speaker.

3.5 User input

The JevonAlarmClock requires user input to switch the light on and off and to configure the alarmclock functionality. To enable this 4 pushbuttons have been added that are directly connected to GPIO pins on the ARM microcontroller. These buttons don't have hardware debouncing, this will be taken care off in the software.

3.6 Schematics



Chapter 4

JevonAlarmClock software

4.1 Tools

The embedded software for the JevonAlarmClock was developed using the Keil μVision 5 software. Additionally Eclipse Luna CDT was used for refactoring and formatting the sourcecode.

4.2 Architecture and design

The embedded software for the JevonAlarmClock uses the following frameworks and libraries:

- CMSIS-CORE
- CMSIS-Driver for UART and SPI
- CMSIS-RTOS
- Segger emWin for the LCD GUI
- STM32F4xx HAL drivers

The use of CMSIS RTOS(Keil RTX) enables the use of threads. The applications has the following user defined threads:

- Main thread
- UART thread
- GUI thread

The entire code is written in C and split into files per functionality.

4.3 serial communication with XBee module

One thread is dedicated to UART communication with the Zigbee XBee module. The UART functionality is build upon the CMSIS UART driver. UART communication is using pins PC10 and PC11 as TX and RX respectively.

4.4 GUI

The GUI is updated using a dedicated GUI thread. The GUI functionality is build using the emWin library. The entire application uses only 1 dialog, in which a number of widgets are displayed. The exacts contents of the GUI depend on the state of the program. There are 3 states defined:

- NORMAL: Default screen
- SETALARM: The screen to set the alarm time
- ALARM: The screen displayed when the alarm goes off.

Chapter 5

Gateway software

Chapter 6

Howto build your own JevonAlarmClock

6.1 Bill of materials

- Raspberry Pi + SD card
- Sparkfun XBee Explorer USB
- 2 x Digi XBee module (exact type doesn't really matter, as long as you have 2 of the same type)
- mini USB cable

6.2 Gateway setup

- Go to http://www.raspberrypi.org and follow the instruction on how to install Raspbian on to the SD card.
- Connect your Raspberry Pi to the Internet using an Ethernet cable and boot it with the new Raspbian installation on the SD card.
- Several packages need to be installed in order to build the JevonAlarmClock gateway software on the Raspberry Pi. Log in to your Raspberry Pi and run the following command.

Chapter 7

Conclusion and recommendations