SWEN 563/CMPE 663/EEEE 663
REAL-TIME & EMBEDDED SYSTEMS
ROCHESTER INSTITUTE OF TECHNOLOGY

PROJECT-1: COUNTING AND GENERATION OF RISING EDGE PULSES WITHIN A HISTOGRAM

Submitted to: Larry Kiser(llkiee@rit.edu)

Submitted by:

Nicolas Delanou (nxd7208@rit.edu)

Sai Pradeep Reddy Bijjam (sxb7079@rit.edu)

D.O.S: 09/14/2016

Areas of Focus

Writing of the code:

- Sai Pradeep Reddy Bijjam: 0%
- Nicolas Delanou: 100%

Report redaction:

- Sai Pradeep Reddy Bijjam: Analysis/Design, Block Diagramm
- Nicolas Delanou: Test Plan, Project Results, Lessons Learned

Analysis / Design

In this project we used the concept of bare metal coding which is very essential to communicate with the microcontroller using C-language. The type of board used is STM32L476VG whose functions are controlled using a high performance Cortex-M4 32-bit RISC core with operating frequency of up to 80 MHZ.
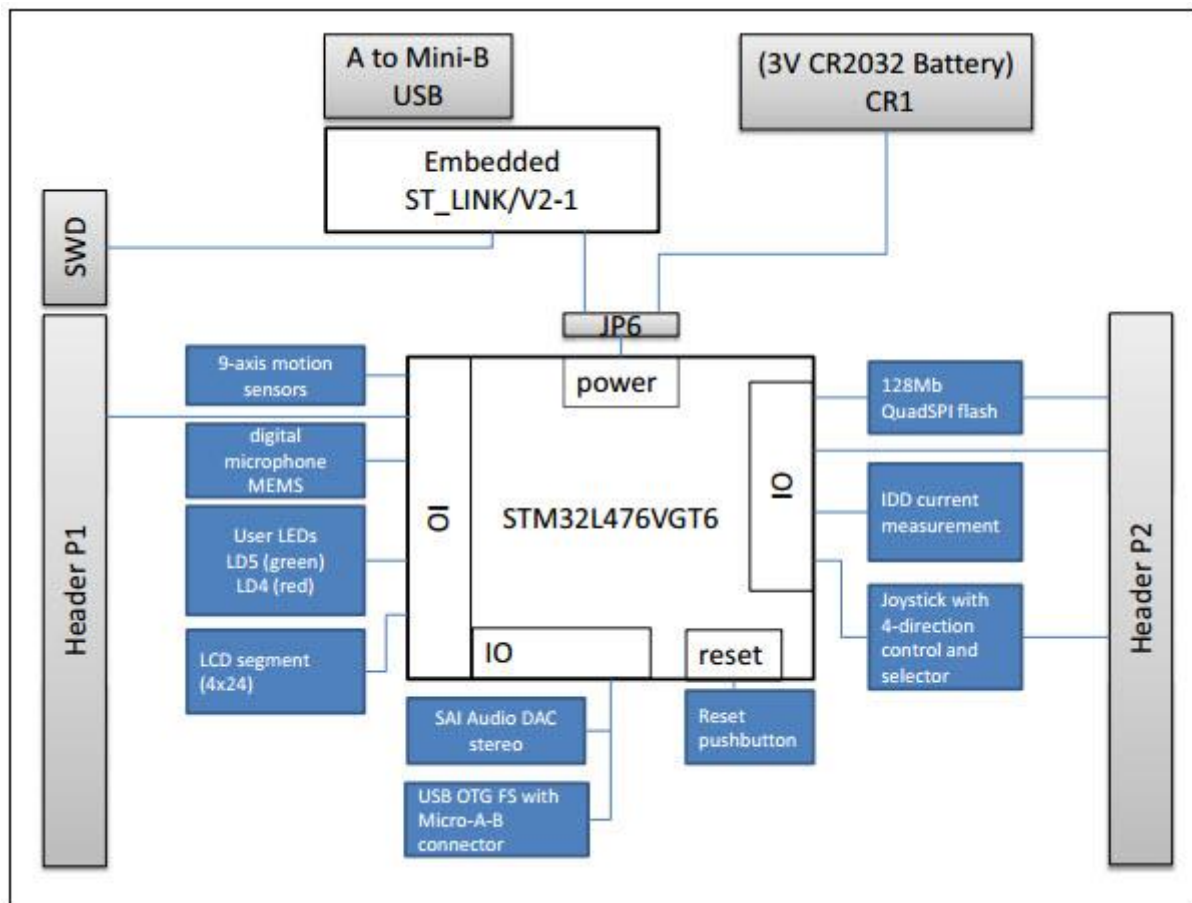
Key features of this device are listed below referring to the manufactures manual:

- Ultra-low-power with Flex Power Control
  - 1.71 V to 3.6 V power supply
  - -40 °C to 85/105/125 °C temperature range
  - 300 nA in VBAT mode: supply for RTC and 32x32-bit backup registers
  - 30 nA Shutdown mode (5 wakeup pins)
  - 120 nA Standby mode (5 wakeup pins)
  - 420 nA Standby mode with RTC
  - 1.1 µA Stop 2 mode, 1.4 µA Stop 2 with RTC
  - 100 µA/MHz run mode
  - Batch acquisition mode (BAM)
  - 4 µs wakeup from Stop mode
  - Brown out reset (BOR) in all modes except shutdown
  - Interconnect matrix
- Core: ARM® 32-bit Cortex®-M4 CPU with FPU, Adaptive real-time accelerator (ART Accelerator™) allowing 0-wait-state execution from Flash memory, frequency up to 80 MHz, MPU, 100DMIPS/1.25DMIPS/MHz (Dhrystone 2.1), and DSP instructions
- Clock Sources
  - 4 to 48 MHz crystal oscillator
  - 32 kHz crystal oscillator for RTC (LSE)
  - Internal 16 MHz factory-trimmed RC (±1%)
  - Internal low-power 32 kHz RC (±5%)
  - Internal multispeed 100 kHz to 48 MHz oscillator, auto-trimmed by LSE (better than ±0.25 % accuracy)
  - 3 PLLs for system clock, USB, audio, ADC
- RTC with HW calendar, alarms and calibration
- LCD 8 × 40 or 4 × 44 with step-up converter
- Up to 24 capacitive sensing channels: support touch key, linear and rotary touch sensors
- 16x timers: 2 x 16-bit advanced motor-control, 2 x 32-bit and 5 x 16-bit general purpose, 2x 16-bit basic, 2x low-power 16-bit timers (available in Stop mode), 2x watchdogs, Sys Tick timer
- Up to 114 fast I/O s, most 5 V-tolerant, up to 14 I/O s with independent supply down to 1.08 V
- Memories
  - Up to 1 MB Flash, 2 banks read-while-write, proprietary code readout protection
  - Up to 128 KB of SRAM including 32 KB with hardware parity check
  - External memory interface for static memories supporting SRAM, PSRAM, NOR and NAND memories
  - Quad SPI memory interface

- 4x digital filters for sigma delta modulator
- Rich analog peripherals (independent supply)
  - 3× 12-bit ADC 5 Msps, up to 16-bit with hardware oversampling, 200 µA/Msps
  - 2x 12-bit DAC, low-power sample and hold
  - 2x operational amplifiers with built-in PGA
  - 2x ultra-low-power comparators
- 18x communication interfaces
  - USB OTG 2.0 full-speed, LPM and BCD
  - 2x SAIs (serial audio interface)
  - 3x I2C FM+(1 Mbit/s), SM Bus/PM Bus
  - 6x USARTs (ISO 7816, LIN, IrDA, modem)
  - 3x SPIs (4x SPIs with the Quad SPI)
  - CAN (2.0B Active) and SDMMC interface
  - SWPMI single wire protocol master I/F
- 14-channel DMA controller
- True random number generator
- CRC calculation unit, 96-bit unique ID
- Development support: serial wire debug (SWD), JTAG, Embedded Trace Macrocell

1. The other hardware components used here are Oscilloscope, Function generator, Male-female wires.
2. The tool used here for coding the logic is Keil -MDK- ARM. This Discovery board had Two advanced timers(TIM1/TIM8), four general purpose timers (TIM2-TIM5) and three extra feature set general purpose timers(TIM15-17) in which each and every timer has 16 memory locations except for the timers 2 & 5 because they provide a wide range having 32 memory locations. Hence, we have picked TIM2 for the "Input Capture Mode".
3. The block diagram of discovery kit consists of 9 -axis motion sensors, Digital microphone MEMS, LEDs, LCD segment,128 mb four directional flash, analog to digital converters, digital to analog converts, Reset push button which is very important while checking the output of the code.
4. The connection mechanism of the above mentioned peripherals can be seen in the block diagram below.

Block Diagram:

STM32L476VGT6 block diagram:

- A to Mini-B USB
- (3V CR2032 Battery) CR1
- Embedded ST_LINK/V2-1
- SWD
- JP6
- power
- Header P1
- 9-axis motion sensors
- digital microphone MEMS
- User LEDs LD5 (green) LD4 (red)
- LCD segment (4x24)
- IO
- STM32L476VGT6
- IO
- IO
- reset
- SAI Audio DAC stereo
- USB OTG FS with Micro-A-B connector
- Reset pushbutton
- 128Mb QuadSPI flash
- IDD current measurement
- Joystick with 4-direction control and selector
- Header P2

Test Plan

The Power On Self Test is the first part of the program and is used to detect if there is an input signal. If this first test is not successful, the program can't continue.

This test control the input of the Timer 2 (*PA0*), used in Capture Mode, during 100 milliseconds and is successful if a rising edge occurs on the pin during this amount of time.

In order to make this working, we are starting the Timer 2 at the beginning of the POST and clear the Capture register. Then we wait for the timer to count to 100ms (using the counter register *TIM2->CNT*). Finally, we watch the capture register (*TIM2->CCR1*). If the register is still blank (0x0), it means that no rising edges have been detected. Otherwise, if the register contains a value and the capture flag is set (*TIM2->SR_CCxIF*), the test is successful.

Here are some screenshots of a failed and a successful POST:

Failed POST

Successful Post

Project Results

At the end, a histogram is displayed on the Serial Terminal. We always get 1000 values in the array out of the 1000 expected. In theory, the histogram should be centered on a value with a few others close to the centered value. The results confirm what we expected, giving us only few different period values.

The results show that the lower the frequency is, the more the results are scattered. Indeed, if a frequency of 1kHz, all the results are in 3 or 4 buckets. With a frequency of 100Hz, the results are spread into more than 10 buckets. With a 10kHz frequency, we only got 2 or 3 different buckets. This fact can be explained by the lack of accuracy of the signal generator.

Here is the result of this tests with 1kHz, 100Hz and 10kHz:

```
Results :
        987 : 120
        988 : 819
        989 : 61
1000 in range out of 1000
Would you like to restart the test ? (y/n)
```

Test with 1kHz

```
Results :
        9968 : 2
        9969 : 4
        9970 : 15
        9971 : 37
        9972 : 75
        9973 : 104
        9974 : 128
        9975 : 154
        9976 : 183
        9977 : 142
        9978 : 84
        9979 : 45
        9980 : 18
        9981 : 4
        9982 : 3
        9984 : 1
        9986 : 1
1000 in range out of 1000
```

Test with 100Hz

```
Results :
        98 : 293
        99 : 707
1000 in range out of 1000
```

Test with 10kHz

Lessons learned

During this project, we had to spend a lot of time locking for information in the Reference Manual and the Datasheet. This was long but this gave us a better understanding of how the board works.

Now, we know how to use the Alternate Functions on the GPIO and use the Timers in Capture Mode. We also used the UART library from the previous lesson.