**/\***

**4357.(008) Embedded Firmware Essentials**

**Spring 2015**

Toggle a GPIO Pin

File: toggle\_gpio.c

Goal: write a code to toggle a GPIO pin as fast as possible.

1. Fast Read LPC178 schematics and data sheets <http://developer.mbed.org/platforms/mbed-LPC1768/>

<http://www.nxp.com/documents/user_manual/UM10360.pdf> (Chapter 9: GPIO)

1. Read ARM Cortex-M3 Datasheet
   1. <http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0337h/CHDDIGAC.html>
   2. <https://ece.uwaterloo.ca/~ece222/ARM/ARM7-TDMI-manual-pt3.pdf>﻿
   3. <http://infocenter.arm.com/help/topic/com.arm.doc.ddi0337h/DDI0337H_cortex_m3_r2p0_trm.pdf>
2. write a C code to toggle a GPIO pin as fast as possible
3. study machine code output from objdump
4. write a page report to tell me the meaning of each bit of the machine code of the while loop body only.

           (you may ignore all machine code before and after the while loop.)

|  |
| --- |
| gcc commands: |
| $ arm-none-eabi-gcc -mcpu=cortex-m3 -mthumb -c hw01.c -o hw01.o |
| $ arm-none-eabi-gcc -mcpu=cortex-m3 -mthumb -O3 -c hw01.c -o hw01\_O3.o |
| $ arm-none-eabi-gcc -mcpu=cortex-m3 -mthumb -Os -c hw01.c -o hw01\_Os.o |
| $ arm-none-eabi-objdump -d hw01.o |

|  |  |  |  |
| --- | --- | --- | --- |
| Xia | Andrew | drew.xia@gmail.com | **mbed: PIN: 5** |
| Varma | Kripa | kripa.varma@gmail.com | **6** |
| Verma | Sandeep | thakursandeep044@gmail.com | **7** |
| Taank | Vilakshan | vilakshantaank@gmail.com | **8** |
| Wu | Takai Kevin | kevinwu1105@hotmail.com | **9** |
| younus | muhammad | myounus@aol.com | **10** |
| Joseph | Anandraj | anand737@yahoo.com | **11** |
| Thomas | Auyeung | thomas0403@yahoo.com | **12** |
| Govindaraju | Abhijit | abhigovindaraju@gmail.com | **13** |
| Patchava | Chaitanya | cpatchava@gmail.com | **14** |
| Fujiwara | Tsuyoshi | tsuyoshifujiwara11@gmail.com | **15** |
| Jayaraman | Anand | anandnayan@gmail.com | **16** |
| Sankara | Bharath | wajsankara@yahoo.com | **17** |
| Hung | Perry | pyhung@gmail.com | **18** |
| Luo | Qihua | qihua578@gmail.com | **19** |
| Rucker | Terri | trrdesigns@comcast.net | **20** |
| Dwayne | Dilbeck | ddilbeck@yahoo.com | **21** |
|  | **paul** | **pcheung347@gmail.com** | **22** |
|  | Eliana | emeridaf@ucsc.edu | **23** |
| Krishna | Ajay | ajaykrishna@gmail.com | **24** |
| Lee | Jason | jasonclee0@gmail.com | **25** |
| ackula | swetha | udayasree23@gmail.com | **26** |
| Olling | Cliff | cliff@bromantech.com | **27** |
| Jerry | Mueckl | jerry.mueckl@gmail.com | **28** |
| Park | Jae Yang | jaeyangp@gmail.com | **29** |
| Frank |  | frank.chao@sbcglobal.net | **30** |

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*VK Code \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include "mbed.h"

int main()

{

uint32\_t volatile \*FIO0DIR0\_VK =(uint32\_t \*)0x2009C001;

uint32\_t volatile \*FIO0SET0\_VK =(uint32\_t \*)0x2009C019;

uint32\_t volatile \*FIO0CLR0\_VK =(uint32\_t \*)0x2009C01D;

uint32\_t volatile \*FIO0MASK0\_VK =(uint32\_t \*)0x2009C011;

// uint8\_t volatile \*FIO0PIN0\_VK = (uint8\_t \*) 0x2009C015;

\*FIO0MASK0\_VK =0xFE; //Set MASK for Port0 GPIO8 or DIP6

\*FIO0DIR0\_VK =0x01; //Set Port0 GPIO8 or DIP6 as Input

\*FIO0CLR0\_VK = 0x01; //Clear Port0 GPIO8 or DIP6 or Set logic low

while(1){

\*FIO0SET0\_VK =0x01; //Set Port0 GPIO8 or DIP6 or Set logic High

//wait(0.5);

\*FIO0CLR0\_VK =0x01; //Clear Port0 GPIO8 or DIP6 or Set logic low

//wait(0.5);

}

}

Report:

I spent 15 hours. The LED blinking was only visible when I added delay.

  The While loop took 5 machine instructions as listed below. The scoping of the signal in oscilloscope show a cycle time 30ns high, 20ns low. The cortex is running at 96MHz and each it takes 10ns to finish 1 Machine Instruction. 5 cycle is 50 ns which matches the observation from scope.

While Loop Machine Code:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| instruction |  |  | OP | register | value |
| 1 | 0x52 | 0x2001 | MOVS | R0 | 1 |
| 2 | 0x54 | 0x6008 | STR | R0 | [R1] |
| 3 | 0x56 | 0x2001 | MOVS | R0 | 1 |
| 4 | 0x58 | 0x6010 | STR | R0 | [R2] |
| 5 | 0x60 | 0xE7FA | B.N | step 56 |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| MOVS | 0x2001 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| STR | 0x6008 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| B.N | 0xE7FA | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |





