**CMPE 240- Advanced Computer Design**

**Assignment:** GPIO Interface Testing

**Name:** Tirumala Saiteja Goruganthu

**SJSU ID:** 016707210

**Team:** Student\_Team1 (Harish Marepalli, Debashish Panigrahi, Shahnawaz Idariya are my team members)

**Professor**: Harry Li

**Date:** 09/20/2022

**Target Board:** LPC1769

**Given Description:**

Import the sample GPIO project to your MCUXpresso IDE, build circuit and modify the C program to realize the following functions:

1. Use p0.21 (j2-23) pin as an output pin and p2.13 (j2-27) pin as an input pin.
2. Build a prototype circuit for GPIO input and output testing. Run GPP testing functions and modify this function to test GPP input and output.
   1. Your output function should be able to turn on LED when CPU sending 1 and turn it off when CPU sending 0.
   2. The input function should be able to read logic 1 when the testing circuit toggle the switch to connect to VCC (3.3v), and logic 0 when the testing circuit switch is toggled to the GND.

**Components Used:**

* + - 1. Prototype Board
      2. LPC1769 inserted into female header pins and soldered to the board.
      3. 330-ohm resistors
      4. Push Button Switch
      5. LED
      6. USB probe to dump the program into on-chip flash memory.

**Source Code:**

/\*

===============================================================================

Name : Assignment\_GPIOTesting.c

Author : Tirumala Saiteja Goruganthu

SJSU ID : 016707210

Course : CMPE240 - Advanced Computer Design

Professor : Harry Li

Description : This program written to test the GPIO functions of LPC1769.

A toggle switch is used to send logic '1' or '0' to CPU where

the input pin (P2.13) will read this logic state and trigger the

output pin (P0.21) accordingly. Note that FIOSET and FIOCLR registers

are used to control the output pin bit and FIOPIN register is used to

control the input pin bit.

===============================================================================

\*/

**#ifdef** \_\_USE\_CMSIS

**#include** "LPC17xx.h"

**#endif**

**#include** <cr\_section\_macros.h>

**#include** <stdio.h>

// **TODO**: insert other include files here

// **TODO**: insert other definitions and declarations here

//Initialize the port and pin as outputs.

**void** **GPIOinitOut**(uint8\_t portNum, uint32\_t pinNum)

{

**if** (portNum == 0)

{

LPC\_GPIO0->FIODIR |= (1 << pinNum);

}

**else** **if** (portNum == 1)

{

LPC\_GPIO1->FIODIR |= (1 << pinNum);

}

**else** **if** (portNum == 2)

{

LPC\_GPIO2->FIODIR |= (1 << pinNum);

}

**else**

{

**puts**("Not a valid port!\n");

}

}

//Initialize the port and pin as inputs.

**void** **GPIOinitIn**(uint8\_t portNum, uint32\_t pinNum)

{

**if** (portNum == 0)

{

LPC\_GPIO0->FIODIR |= (0 << pinNum);

}

**else** **if** (portNum == 1)

{

LPC\_GPIO1->FIODIR |= (0 << pinNum);

}

**else** **if** (portNum == 2)

{

LPC\_GPIO2->FIODIR |= (0 << pinNum);

}

**else**

{

**puts**("Not a valid port!\n");

}

}

//Activate the pin

**void** **setGPIO**(uint8\_t portNum, uint32\_t pinNum)

{

**if** (portNum == 0)

{

LPC\_GPIO0->FIOSET = (1 << pinNum);

**printf**("Pin 0.%d has been set.\n",pinNum);

}

**else**

{

**puts**("Only port 0 is used, try again!\n");

}

}

//Deactivate the pin

**void** **clearGPIO**(uint8\_t portNum, uint32\_t pinNum)

{

**if** (portNum == 0)

{

LPC\_GPIO0->FIOCLR = (1 << pinNum);

**printf**("Pin 0.%d has been cleared.\n", pinNum);

}

**else**

{

**puts**("Only port 0 is used, try again!\n");

}

}

**int** **main**(**void**)

{

//declare switch status variable and pin number

uint32\_t switchstatus;

uint32\_t switchpinnumber = 13;

//Set pin 0.21 as output

GPIOinitOut(0,21);

//Set pin 2.31 as input

GPIOinitIn(2, 13);

**while**(1)

{

//Get the switch status using FIOPIN Register

switchstatus = (LPC\_GPIO2->FIOPIN >> switchpinnumber) & 0x01;

//Since all pins that are configured as Input will be in high state

//due to the internal pull-ups which is why switch status is toggled

//using logic '0'

**if** (switchstatus == 1)

{

//Activate pin 0.21

setGPIO(0,21);

}

**else**

{

//Deactivate pin 2.13

clearGPIO(0, 21);

}

}

//0 should never be returned, due to infinite while loop

**return** 0;

}

**Connectivity Table:**

|  |  |  |
| --- | --- | --- |
| **CPU PIN** | **J2-TAG PIN** | **FUNCTION** |
| P0.21 | J2-23 | Output |
| P2.13 | J2-27 | Input |

**Schematic Diagram:**

Diagram, schematic

Description automatically generated

**Output Snippet:**

Below is the snippet of the output with LED in ON state. User will send logic-1 to the input pin of LPC1769 by pushing the button-in. This logic switch at the input pin is realized by the above code and toggles the logic value at the output port accordingly which in-turn will light the LED.

A hand holding a circuit board

Description automatically generated with low confidence

**Conclusion:**

Hence, modified the given C program and tested the GPIO interface in LPC1769 using the pins P0.21 for output and P2.13 for input on a prototype board by building the circuit.