/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*Filename : readmcp300x.c

\*Description : read ADC mcp3008 value with raspberry pi

\*Company : SunRobotics Technologies

\*Website : www.sunrobotics.co.in

\*E-mail : support@sunrobotics.co.in(For Any Query)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//#define \_GNU\_SOURCE

#include <unistd.h>

#include <stdint.h>

#include <string.h>

#include <errno.h>

#include <wiringPi.h>

#include <stdio.h>

#include <stdlib.h>

#include <wiringPiSPI.h>

#define TRUE (1==1)

#define FALSE (!TRUE)

#define CHAN\_CONFIG\_SINGLE 8

#define CHAN\_CONFIG\_DIFF 0

static int myFd ;

char \*usage = "Usage: mcp3008 all|analogChannel[1-8] [-l] [-ce1] [-d]";

// -l = load SPI driver, default: do not load

// -ce1 = spi analogChannel 1, default: 0

// -d = differential analogChannel input, default: single ended

void loadSpiDriver()

{

if (system("gpio load spi") == -1)

{

fprintf (stderr, "Can't load the SPI driver: %s\n", strerror (errno)) ;

exit (EXIT\_FAILURE) ;

}

}

void spiSetup (int spiChannel)

{

if ((myFd = wiringPiSPISetup (spiChannel, 10000)) < 0)

{

fprintf (stderr, "Can't open the SPI bus: %s\n", strerror (errno)) ;

exit (EXIT\_FAILURE) ;

}

}

int myAnalogRead(int spiChannel,int channelConfig,int analogChannel)

{

if(analogChannel<0 || analogChannel>7)

return -1;

unsigned char buffer[3] = {1}; // start bit

buffer[1] = (channelConfig+analogChannel) << 4;

wiringPiSPIDataRW(spiChannel, buffer, 3);

return ( (buffer[1] & 3 ) << 8 ) + buffer[2]; // get last 10 bits

}

void print\_info()

{

printf("\n");

printf("|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |\n");

printf("| Read MCP3008(3004) ADC value |\n");

printf("| ------------------------------ |\n");

printf("| | ADC | | Pi | |\n");

printf("| |-----|-----------|-----| |\n");

printf("| | CS | connect to| CE0 | |\n");

printf("| | Din | connect to| MOSI| |\n");

printf("| | Dout| connect to| MISO| |\n");

printf("| | CLK | connect to| SCLK| |\n");

printf("| | CH0 | connect to| 3.3V| |\n");

printf("| | CH1 | connect to| GND | |\n");

printf("| |\n");

printf("|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\n");

printf("\n");

}

int main (int argc, char \*argv [])

{

int loadSpi=FALSE;

int analogChannel=0;

int spiChannel=0;

int channelConfig=CHAN\_CONFIG\_SINGLE;

if (argc < 2)

{

fprintf (stderr, "%s\n", usage) ;

return 1 ;

}

if((strcasecmp (argv [1], "all") == 0) )

argv[1] = "0";

if ( (sscanf (argv[1], "%i", &analogChannel)!=1) || analogChannel < 0 || analogChannel > 8 )

{

printf ("%s\n", usage) ;

return 1 ;

}

int i;

for(i=2; i<argc; i++)

{

if (strcasecmp (argv [i], "-l") == 0 || strcasecmp (argv [i], "-load") == 0)

loadSpi=TRUE;

else if (strcasecmp (argv [i], "-ce1") == 0)

spiChannel=1;

else if (strcasecmp (argv [i], "-d") == 0 || strcasecmp (argv [i], "-diff") == 0)

channelConfig=CHAN\_CONFIG\_DIFF;

}

//

if(loadSpi==TRUE)

loadSpiDriver();

wiringPiSetup () ;

spiSetup(spiChannel);

print\_info();

//

if(analogChannel>0)

{

printf("MCP3008(CE%d,%s): analogChannel %d = %d\n",spiChannel,(channelConfig==CHAN\_CONFIG\_SINGLE)

?"single-ended":"differential",analogChannel,myAnalogRead(spiChannel,channelConfig,analogChannel-1));

}

else

{

for(i=0; i<8; i++)

{

printf("MCP3008(CE%d,%s): analogChannel %d = %d\n",spiChannel,(channelConfig==CHAN\_CONFIG\_SINGLE)

?"single-ended":"differential",i+1,myAnalogRead(spiChannel,channelConfig,i));

}

}

close (myFd) ;

return 0;

}