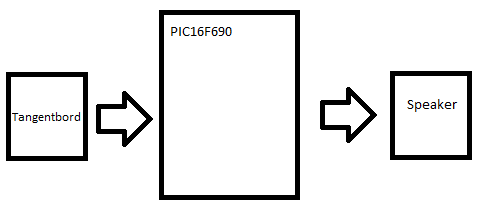
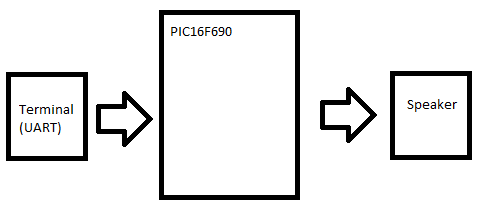
|  |
| --- |
| Kungliga Tekniska Högskolan |
| Pic16F690 laboration synt |
|  |
|  |
| **Sadok Habibi Dalin, Alexander Barosen** |
| **2014-05-09**  **Sadokd@kth.se, Abarosen@kth.se**  **Inbyggd elektronik IE1206** |
|  |

# Beskrivning av produktidén



Idén var att spela upp ljud i högtalaren genom att mata in tecken i Pic-processorn med tangentbordet. Processorn skulle spela olika toner beroende på vilket tecken som matades in. Flera knappar skulle kunna tryckas ner samtidigt för att generera ett ackord. Målet var att kunna spela 32 olika toner för att modellera en mindre synt. Långa toner skulle kunna spelas upp utan avbrott eller fördröjningar.

# Beskrivning av funktionsprototypen

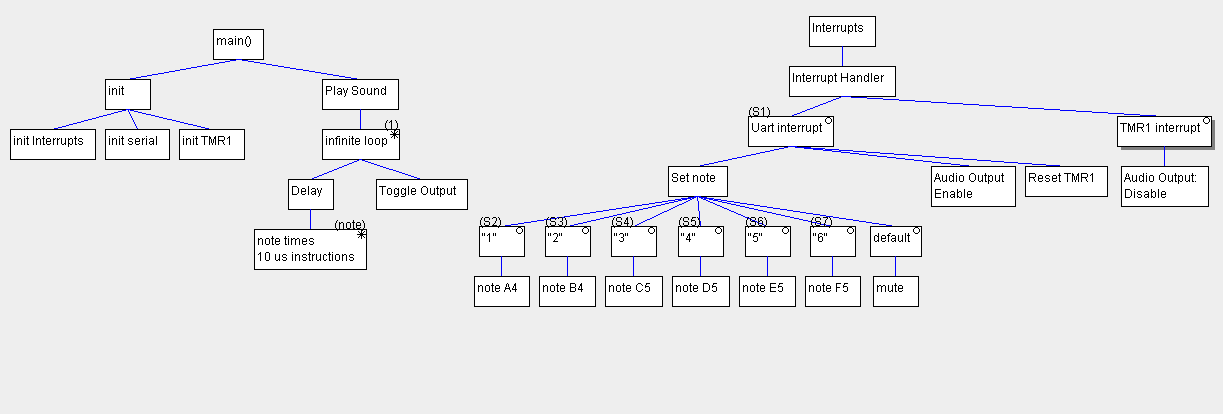


Funktionsprototypen består av två delar, en Pic-processor och ett högtalarelement. Pic-processorn styrs av datorns tangentbord eller liknande inmatningsverktyg. Inmatningsformatet är av typen ASCII och tas in via UART in i processorn. Processorn tolkar sedan ASCII-koden och spelar en ton ut genom högtalaren. Olika toner spelas beroende på vilken knapp som trycks ned på tangentbordet. Prototypen kan hittills tolka 6 olika tangentbordsknappar. Därför kan vi bara spela 6 olika toner. Det går ej att spela upp ackord och långa toner spelas upp som många korta toner i serie.

# Checklista, kontroll av program

* Processorn ska kunna ta emot tecken från ett tangentbord. Detta testades genom att ta emot tecken genom UART och sedan skicka tillbaka mottagna tecknet till sändaren.
* Ljud ska kunna spelas upp genom högtalaren. Detta testades med färdig kod från mel.c[[1]](#footnote-1).
* Processorn ska kunna tolka olika tecken från tangentbordet och spela upp olika ljud beroende på tecknet. Detta testades genom att köra programmet, trycka på olika tangentbordsknappar och lyssna.
* Ljud ska spelas upp när en knapp släpps. Detta kontrollerades genom att ha en timer som stänger av ljudet när knappen släpps.
* Flera noter ska kunna spelas i snabb följd utan avbrott eller fördröjningar. Detta testades genom att spela olika noter i en snabb följd.
* Långa noter ska kunna spelas genom att hålla in en knapp utan avbrott. Detta kontrollerades genom att hålla in en knapp länge.
* Inga knappar än de programmerade ska ha någon inverkan på programmet. Detta testades genom att trycka på alla andra knappar än de programmerade och lyssna efter ljud. Kombinationer av tillåtna och otillåtna knappar testades också för att utesluta övriga eventuella fel.

# Strukturdiagram



# Programkod

/\* B Knudsen Cc5x C-compiler - not ANSI-C

Authors: Sadok Habibi Dalin, Alexander Barosen.

This program plays different notes by taking input from the uart and translating it into

different pitches. The program uses interrupts to change the current note and a timer to

control note length.

\*/

#include "16F690.h"

#include "int16Cxx.h"

#pragma config |= 0x00D4

//Global variables.

char note; //Used to choose note to play.

unsigned long T1val; //TMR1 start value.

//Function declarations.

void initInt( void );

void initserial( void );

void initTMR1( void );

#pragma origin 4 //Put interrupt function here.

interrupt intHandler(void)

{

int\_save\_registers

if(RCIF == 1) //Uart trigger flag, triggers on note input.

{

switch (RCREG)//comparison of input.

{

case '1':

note = 114; //A4

break;

case '2':

note = 101; //B4 break;

case '3':

note = 96; //C5

break;

case '4':

note = 85; //D5

break;

case '5':

note = 76; //E5

break;

case '6':

note = 72; //F5

break;

default:

note = 0; //Mute note

}

TRISA.4 = 0; //Enable audio output

TMR1L = T1val % 256; //Reset TMR1.

TMR1H = T1val / 256;

}

if(TMR1IF == 1) //TMR1 Interrupt flag.

{

TRISA.4 = 1; //Disable audio output.

TMR1IF = 0; //Acknowledge interrupt.

}

int\_restore\_registers

}

void main(void)

{

initInt(); //Initialize interrupts.

initserial(); //Initialize inbound serialport(Uart).

initTMR1(); //Initialize TMR1.

bit out;

/\* speaker output bit. BORROWED CODE(mel.c, [www.ict.kth.se/courses/IE1206/project/mini/mel/index.htm](http://www.ict.kth.se/courses/IE1206/project/mini/mel/index.htm)). \*/

/\*

The infinite while-loop contains code that will play a note forever by turning the PORTA.4 on and off.

The pitch of the sound is dependant on the delay between changes of PORTA.4.

The length of the for-loop that causes the delay is dependant on the note variable.

The original code has been modified to ouput a sound indefinitely and is instead controlled by the interrupt routine.

\*/

//BORROWED CODE START MARKER

/\* mel.c, <http://www.ict.kth.se/courses/IE1206/project/mini/mel/index.htm> \*/

while (1)

{

char j;

for(j = note; j > 0; j--)

{

nop(); nop(); nop(); nop();

}

/\* Toggle Output bit RA4 On/Off \*/

out = !out;

PORTA.4 = out;

}

//BORROWED CODE END MARKER

}

//Functions below.

void initInt(void)

{

GIE = 1; //Global interrupt enable bit.

RCIE = 1; //Uart receive interrupt enable bit.

PEIE = 1; //Peripheral interrupt enable bit.

}

void initserial(void) //initialise PIC16F690 inbound serial port

{

//One start bit, one stop bit, 8 data bit, no parity. 9600 Baud.

SYNC = 0; //asynchronous operation

SPEN = 1; //serial port enable.

BRG16 = 1; //4 MHz clock frequency

SPBRG = 25; //baud rate timer period

CREN = 1; //Continuous receive

RX9 = 0; //8 bit reception

ANSELH.3 = 0; //RB5 not AD-input but serial\_in

TRISB.7 = 1; //Rx input

}

void initTMR1(void)

{

T1val = 600; //2^16 - 601 ticks before overflow.

TMR1L = T1val % 256; //Reset TMR1.

TMR1H = T1val / 256;

T1CKPS1 = 1; //Timer1 Prescaler.

T1CKPS0 = 1; //Need slower timer for longer notes.

TMR1IE = 1; //TMR1 overflow interrupt enable bit.

TMR1ON = 1; //TMR1 enable bit.

}

//Hardware.

/\*

\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

| \/ |

+5V---|Vdd 16F690 Vss|---Gnd

|RA5 RA0/AN0/(PGD)|

|RA4/AN3 RA1/(PGC)|

Speaker<--|RA3/!MCLR/(Vpp) RA2/AN2/INT|

|RC5/CCP RC0|

|RC4 RC1|

|RC3/AN7 RC2|

|RC6/AN8 AN10/RB4|

|RC7/AN9 AN11/RB5/Rx|<--Uart input

not used|RB7/Tx RB6|

|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|

\*/

1. http://www.ict.kth.se/courses/IE1206/project/mini/mel/index.htm [↑](#footnote-ref-1)