Projektrapport

EITF40 – Digitala projekt
Handledare: Bertil Lindvall

Johan Wennersten (dt08jw2@student.lth.se)
Arvid Lindell (dt08al6@student.lth.se)

Abstract

This report describes relevant technical details in a project for the course EITF40 at Lunds tekniska högskola. The primary purpose of the project was to construct an embedded system which could be used as a primitive synthesizer, operated by an ordinary PS/2-based keyboard. Another purpose was to provide conversion of PS/2 data into USB HID, which enables usage of older keyboards on newer PCs.

Innehåll

1 Introduktion

I mitten av tvåan skrev projektdeltagarna ett program i Java vars enda syfte var att producera ljud av olika frekvenser beroende på vilken knapp på tangentbordet som trycktes ner. Det valda tillvägagångssättet hade diverse problem och tillät exempelvis inte att programmet låg i bakgrunden och lät när man använde datorn till annat.

Då den funktion programmet erbjöd var så pass produktivitetshöjande som den var bestämde sig deltagarna för att återskapa funktionaliteten i ett inbyggt system, liggandes mellan tangentbordet och datorn.

Vidare bestämdes det att konstruktionen även skulle kunna omvandla mottagna PS/2 signaler till USB HID.

2 Kravspecifikation

2.1 Initiell kravspecifikation

2.1.1 Hårdvarukrav

- Konstruktionen ska ha en 128x64 pixlars display.
- Konstruktionen ska kunna tolka signaler från ett tangentbord (AT/PS2-standard) och skicka dessa vidare till en ansluten dator (antingen via USB eller AT/PS2).
- Konstruktionen ska kunna producera ljud med hjälp av en D/A-omvandlare via en inbyggd högtalare (eventuell anslutning via 3,5 millimeters jack kan vara aktuellt, för anslutning av extern högtalare). En förstärkare kan komma att behövas.
- Konstruktionen ska eventuellt kunna kommunicera med en ansluten dator via RS232.

2.1.2 Funktionsmässiga krav

- Konstruktionen ska kunna samla och visa statistik angående tangentbordsanvändningen (exempelvis antal ord/nedslag per sekund/minut). Denna statistik ska kunna visas upp på displayen.
- Konstruktionen ska kunna spela upp olika sorters toner (vanliga sinus el. exempelvis fyrkantsvågor) baserat på tangentnedslag. Tangentbordet ska med andra ord i samverkan med resten av systemet kunna användas som en slags synthesizer.
- Konstruktionen ska kunna analysera längre strängar av inmatad data, och ge feedback till användaren via displayen eller högtalaren. Exempelvis kan en lista på vanliga stavfel ligga lagrade i en lista, om något av dessa matas in kommer systemet att automatiskt korrigera inmatningen eller notifiera användaren, via exempelvis ett bröl från högtalaren.
- Styrsignaler ska kunna skickas från tangentbordet till systemet som slår av eller på olika inbyggda funktioner (exempelvis stänga av ljud, eller byta till en annan tonuppsättning).
- Felmeddelanden ska kunna visas på displayen.

2.2 Avvikelser

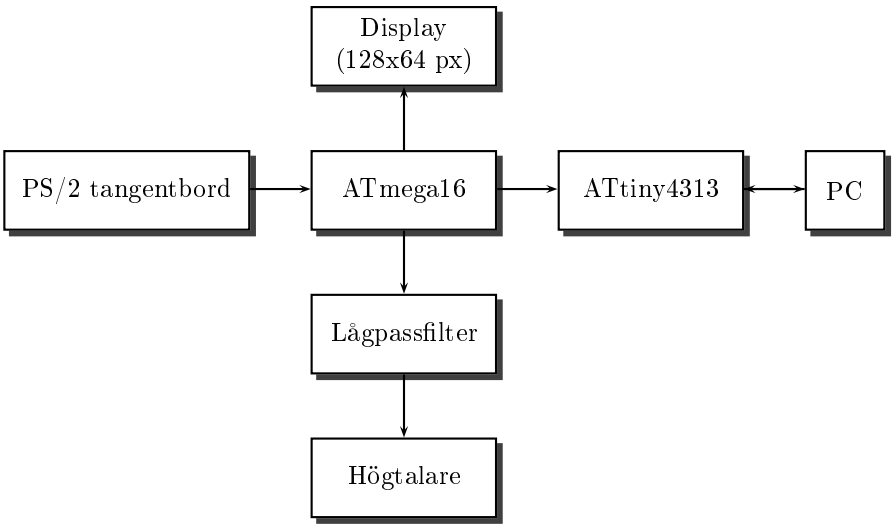
De krav vi inte implementerat är att konstruktionen ska kunna analysera längre strängar av inmatad data, och ge feedback till användaren via displayen eller högtalaren, samt att statistik över tangentbordsanvändningen ska kunna visas upp.

Gällande hårdvaran slopade vi D/A omvandlaren och använder istället PWM-funktionaliteten hos konstruktionens huvudprocessor i samverkan med ett lågpassfilter (se sektion 3.1.3). En ytterligare avvikelse är att vi använder en extern, och förstärkt högtalare istället för en inbyggd.

3 Konstruktionen

Konstruktionen omvandlar mottagen PS/2 data till USB HID, detta innebär att man kan använda äldre tangentbord på datorsystem som exempelvis inte har en PS/2 port. Den går även att använda som primitiv synthesizer, med toner mappade till tangenter. I vissa sammanhang kan den även anses vara produktivitetshöjande, då användaren får en hårdvarubaserad auditoriell feedback genom sin inmatning som i vanliga fall ej kan uppnås utan exempelvis extra mjukvara.

Själva huvudprocessorn består av en ATmega16, omvandlingen från AT/PS2 till USB HID sker i en intilliggande ATtiny4313. Vidare används en display och en högtalare.



Figur 1: Blockschema över konstruktionen

3.1 ATmega16

3.1.1 Display

ATmega16 ansvarar för att styra displayen (128x64 px), som i nuläget visar vågformen av det producerade ljudet samt gällande inställningar.

3.1.2 Inläsning från PS/2

mega16 ansvarar även för att läsa in nedtryckta knappar från ett inkopplat PS/2-tangentbord. Dessa läggs i en lista och skickas vidare till både pianofunktionaliteten såväl som ATtiny4313 (via USART, för USB HID-funktionalitet).

Mottagningen sker genom att tangentbordet först skickar en startbit, följt av 8 databitar, en paritetsbit och stoppbit. Inläsningen sker interruptbaserat (PS/2 clock, nedåtgående flank) enligt följande kodstycke[1].

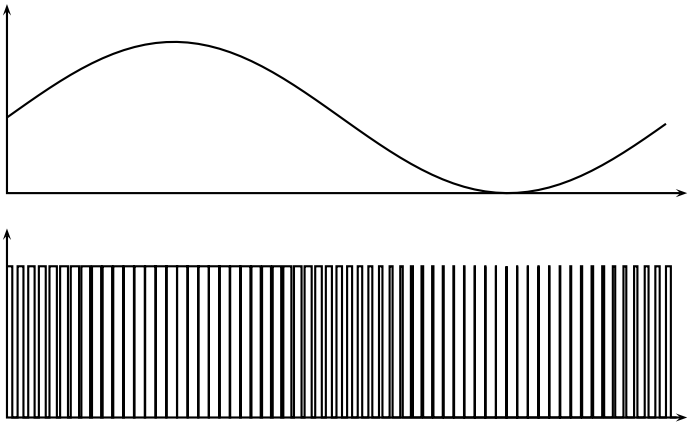
[1] En ingående beskrivning av protokollet kan återfinnas på `http://www.`

```
ISR (INT1_vect)
{
    interrupt++;

    if (interrupt == 1) {
        start_bit = RB(D_DATA_PIN, D_DATA);
    } else if (interrupt < 10) {
        WB(scancode, interrupt - 2, RB(D_DATA_PIN, D_DATA));
    } else if (interrupt == 10) {
        parity_bit = RB(D_DATA_PIN, D_DATA);
    } else if (interrupt == 11) {
        stop_bit = RB(D_DATA_PIN, D_DATA);
        interrupt = 0;
        buff_put(&scancodes, scancode);
    }
}
```

3.1.3 Ljud

Ett egenhändigt tillverkat MATLAB-skript producerar sinusvågor baserat på pianofrekvenser och vald uppspelningstakt. Dessa lagras sedan som sampel i mega16, och sätts ihop till ett slutgiltigt resultat (baserat på nedtryckta tangenter) som sedan efterbehandlas (med exempelvis LFO[2]) innan det skrivs till en 8-bitars PWM-port.



Figur 2: Analog och PWM-baserad sinuskurva

Signalen filtreras genom ett lågpassfilter för att plocka bort oönskade PWM-frekvenser[3], för att få en snarlik analog signal (se fig 2). Så vitt vi kan förstå sker detta med hjälp av magi. Den filtrerade signalen kopplas sedan till en förstärkt högtalare.

I nuläget stöder ljudfunktionaliteten två lägen, dels ett vanligt, där utvalda tangenter spelar upp en enda pianofrekvens. Tangenterna är utlagda på samma

computer-engineering.org/ps2protocol.

[2] http://en.wikipedia.org/wiki/Low-frequency_oscillation

[3] http://ltwiki.org/images/8/82/PWM_Filters.pdf

sätt som de vore på ett vanligt piano. Det andra läget har vi valt att kalla för kaosläget, och med detta läge aktiverat slumpas det fram en ny ton vid varje tangentnedslag.

3.1.4 Main

För tydlighets skull beskrivs huvudprocessorns mainloop nedan.

```
medans för alltid

    cykelcount = 0
    rendera ljudet och skriv till pwm

    om några tangenter tryckts ner eller släppts upp
        skicka dessa till tiny4313
        uppdatera piano-funktionaliteten

    skriv saker till displayen

    medans cykelcount < antalet för 22050 hz
        vänta
```

3.2 ATtiny4313

3.2.1 USB HID

Som tidigare nämnt ansvarar tiny4313 för att göra om PS/2 till USB HID. mega16 och tiny4313 kommunicerar via USART. mega16 skickar de emottagna PS/2-signalerna till tiny4313 som behandlar dessa och med hjälp av det färdigskrivna biblioteket VUSB[4] utförs PS/2 till USB omvandling.

Vi har valt att implementera delar av standarden, och stöder exempelvis inte bootprotokollet[5] eller uppdatering av tangentbordets lysdioder.

Omvandlingen utförs i princip med hjälp av en lookup-table och funktionalitet för att ta bort och lägga till tangenter i en behövlig USB-rapport. Koden är relativt lättläst och den intresserade läsaren refereras till denna för detaljer.

4 Problem och implementationsmässigt plur

4.1 Uppspelningsfrekvens

Själva ljuduppspelningstakten är på 22050 hz, vilket motsvarar radiokvalitet. Vi hade först tänkt nöja oss med 12070 hz, men det visade sig att vi hade mängder med både klockcykler och plats över i flashminnet för större och mer detaljerade sampel.

4.2 Displayen

Vi fann att displayens styrkrets var långsam i förhållande till resten av systemet. Om kommandon skickades snabbare än ett per 11:e mikrosekund uppstod

[4] `http://www.obdev.at/products/vusb/index.html`

[5] Möjliggör användning när ingen USB-drivrutin finns laddad, exempelvis i BIOS.

det troligen bitfel och följdaktligen trasig bild. För att bibehålla en konsekvent ljuduppspelning valde vi istället att skicka ett kommando till displayen varje iteration av mainloopen. Vidare optimering, för att ge en mer responsiv plot av ljudsignalen, sveper vi över displayen och ändrar bara de pixlar som behöver uppdateras, och nollställer dessa vid nästa iteration. Alternativet till detta hade varit att uppdatera all bildyta vid varje iteration.

4.3 Störningar

Störningar uppstod när fler och fler komponenter började baxas på brädan. Dessa störde ut USB-funktionaliteten. Vi vet inte var dessa kom ifrån, och vi lyckades inte bli av med dem på ett elegant sätt (vi flyttade bara bort tiny4313 från kortet till en avlägsen breadboard). Vi provade dels att koppla upp ett jordplan samt att avkoppla jord och 5v på både mega16 och tiny4313 med vars en 100nF kondensator.

4.4 VUSB

VUSB är en mjukvarubaserad och bit-bangad[6] drivrutin, vilket innebär att den slukar ganska mycket CPU-tid. USB har väldiga krav på timing, detta sammankopplat med det förstnämnda faktumet leder till att det kan bli lite trixigt att utveckla timingkritiska applikationer (som exempelvis PS/2-kommunikation). Då vi använder tiny4313s befintliga USART för kommunikation med mega16 var en första lösning att använda en mjukvarubaserad USART för debugutskrifter. Med denna blev felsökning problematiskt då den mjukvarubaserade USARTen var långsam i förhållande till resten av systemet (9600 baud utan allvarligare bitfel). Resultatet av detta var att vi förlorade inkommande data, trots att mottagningen är interruptdriven.

För att komma runt denna problematik dedikerade vi en port på tiny4313 som skriver ett 8-bitars värde till en tiny2313, denna lagrar detta, och skriver i tur och ordning ut de mottagna värdena på sin egen USART (som är kopplad till en PC).

[6] `http://en.wikipedia.org/wiki/Bit_banging`

5 Resultat

Slutresultatet blev ganska bra. Störningarna är väldigt irriterande, men oftast fungerar konstruktionen bra ändå. Vi gjorde inte riktigt som vi hade tänkt först, men vi anser att de avvägningar vi har gjort är rimliga, och har lett till ett enklare system med samma funktionalitet.

Delar av denna rapport är skriven med hjälp av konstruktionen, vilket har lett till en klart ökad produktivitet i författandet[7]. Med kaosläget aktiverat blir det ett rent nöje att skriva stycken av högt innehållsmässigt värde som det som står härnedan.

eöy -p aoeu det är en gnaska bra konstruktion ändå får man säga faktiskt BLIPP BLIPP BLIPP BLIPP BLIPP BLIPP BLIPP BLIPP BLIPP BLIPP BLIPP IBLPP BLIPP BLIPP BLIUHhuhu

[7] Och en del sinnessjuka.