

Berrastation

Digitala Projekt EITF11

Lunds Universitet

2016-04-20

Oscar Gustavsson 930325-5492, Daniel Wisenhoff 9310116232

Inledning

BerraStation är slutresultatet av projektet i kursen Digitala Projekt (EITF11) vid LTH. Syftet med kursen är att förstå konstruktionsprocessen i digitala projekt och att ta fram en prototyp av valfri produkt. Projektet ger insikt i både hur man använder hårdvara såväl som mjukvara.

Innehåll

[Berrastation](#)

[Inledning](#)

[Innehåll](#)

[Kravspecifikation](#)

[Kopplingsschema](#)

[Hårdvara](#)

[Processor](#)

[FTDI-chip](#)

[Temperatursensor](#)

[Resistorer](#)

[Kondensator](#)

[Mjukvara](#)

[C](#)

[Java](#)

[JTAG](#)

[Genomförande](#)

[Idégenerering](#)

[Planering](#)

[Bygga och kodning](#)

[Resultat](#)

[Väderstationen](#)

[Javaprogrammet](#)

[Diskussion](#)

[Referenser](#)

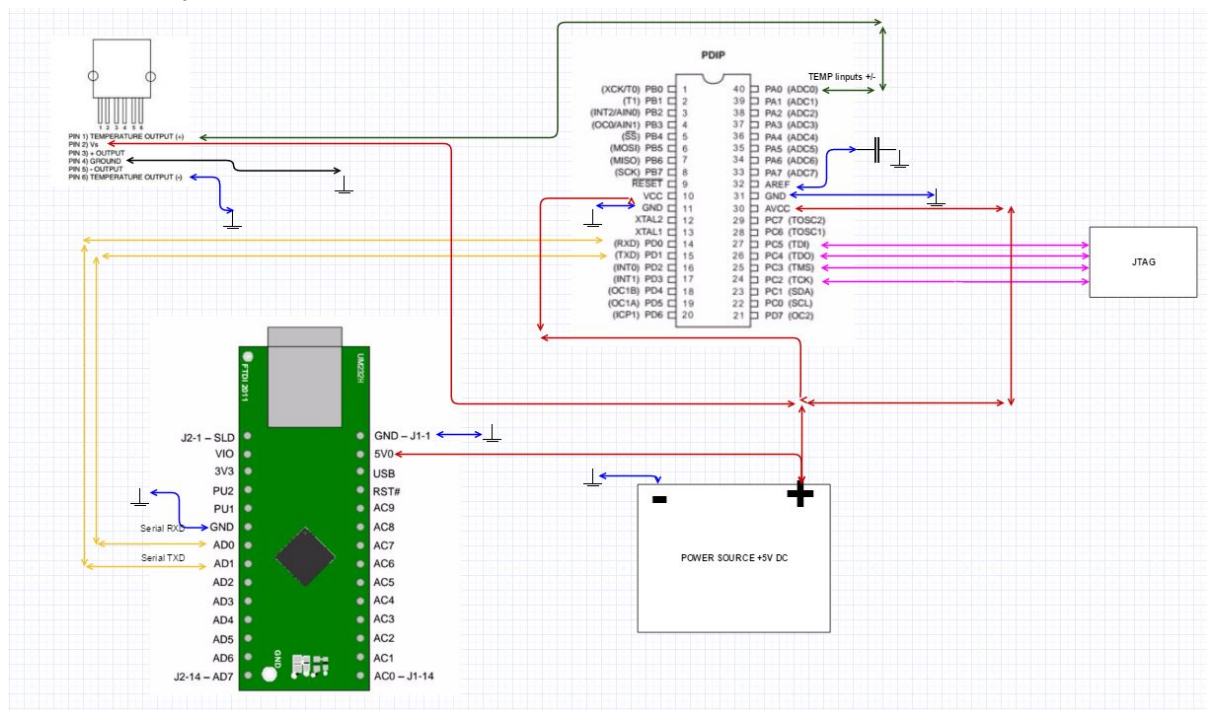
Kravspecifikation

Väderstationen ska:

- Mäta temperatur
- Skicka data över USB till en dator
- Presentera data i applikation grafiskt
- Ha en LED-lampa som indikerar om data skickas

Kopplingsschema

Kopplingsschemat har uppdaterats under arbetets gång, det är således inte rekommenderat att ta alltför mycket inspiration av bilden nedan.



Hårvara

Processor

Processorn som används är en ATmega16 processor med 8 MHz klockfrekvens från Atmel. Processorn programmeras med hjälp av en JTAG kopplad till datorn. ATmega16 har 32 stycken pinnar och 16 kb minne.

FTDI-chip

Ett FTDI-chip användes för överföring av data från väderstationen till PC:n där informationen presenteras. Detta sker med hjälp av en USB-kabel från chippet till datorn.

Temperatursensor

Sensorn som användes var en LM335 som mäter temperaturer mellan -40 grader Celsius och 100 grader Celsius med en exakthet på +/- 1 grad Celsius. En grads ökning motsvarar 10mV spänningsskillnad och genom detta förhållande kan en temperatur fås.

Resistorer

Resistorer av olika storlek användes mellan FTDI, temperatursensor, lampa och processor.

Kondensator

En kondensator användes för att stabilisera spänningen över processorn.

Mjukvara

C

Koden till ATmega16 skrevs i programmeringsspråket C. Huvudsyftet var att få vidare informationen från temperatursensorn via FTDI till datorn där data sedan presenteras. Beräkningstunga delar lades i Java-programmet då datorn har mer processorkraft. En status-led programmerades också vars syfte var att blinka varje gång information skickades. AtmelStudio användes för all C-programmering.

Java

JavaFX används för att presentera temperaturen i ett diagram som uppdateras varje gång data har mottagits från BerraStation. Två klasser skapades, en klass som läser in data och en Main-klass som presenterar data i diagrammet. Eclipse användes för all Java-programmering.

För att få Java att "prata" med serial-porten behövdes ett externt bibliotek implementeras. jSSC användes för detta syfte. Vi rekommenderar ej andra grupper att använda Java i fortsättningen. Istället är en mjukvara gjord i Visual Studio och C/C++/C# ett mycket enklare alternativ.

JTAG

JTAG användes för att föra över C-koden till BerraStation.

Genomförande

Idégenerering

Idén kom efter vår önskan att göra något som kan användas i det vardagliga livet samtidigt som vi ville göra något som ingen annan grupp hade gjort tidigare. Valet föll därför naturligtvis på att göra en väderstation, anno 2016. Med en anslutning via USB som presenterar data på ett modernt medium - datorn.

Planering

Innan kursstart visades en enkel kravspecifikation upp för handledare som godkände projektet. Lämpliga komponenter och tillhörande datablad valdes. Då inga liknande projekt hade genomförts tidigare vad målet att komma igång snabbt så att det skulle finnas tid för att lösa eventuella svårigheter.

Första steget var att göra ett kopplingsschema, se rubrik Kopplingsschema. Med hjälp av databladen kunde ett lämpligt kopplingsschema skapas som sedan handledaren godkände.

Bygge och kodning

Kodning och bygge skedde parallellt, både för att säkerhetsställa att komponenterna var rätt kopplade, samt att koden var rätt skriven. Innan temperaturmätaren kopplades in gjordes ett testprogram i C för se så att data kunde skickas och tas emot på ett korrekt sätt. En LED-lampa kopplades också in som visades om data skickades från processorn eller ej.

Resultat

Den slutgiltiga prototypen består huvudsakligen av två delar; väderstationen samt javaprogrammet på en PC. Nedan beskrivs de två delarna separat.

Väderstationen

Väderstationen tar emot data två gånger per sekund och skickar varje sekund över mätningarna till Javaprogrammet via FTDI-chippet. En status-LED blinkar varje gång information förs över till ansluten PC. LM335 chippet klarar av att mäta med en noggrannhet på $\pm 0,5$ grader Celcius och informationen skickas via USB till värddatorn.

Javaprogrammet

Javaprogrammet tar varje sekund emot två stycken värden från väderstationen. Värdena slås samman för att få ett medelvärde som sedan presenteras i en graf på PC:n. Grafen

uppdateras varje sekund och data hämtas så länge programmet körs och väderstationen är inkopplad. Programmet känner även av om något värde är orimligt eller mottages felaktigt. Skulle detta uppstå bortser programmet från detta och inväntar nästa mätvärde istället.

Diskussion

Vi hade tidigare inte gjort något liknande projekt och därför var våra kunskaper väldigt begränsade. När vi började ville vi lära oss så mycket som möjligt, men samtidigt inte sätta orimliga mål. Det mest tidskrävande delen av projektet var att faktiskt sälla information från databladen och förstå vad som var relevant för vårt projekt. All information fanns framför oss, men med begränsad kunskap var det svårt att förstå vad som vi faktiskt skulle använda.

Skulle projektet gjorts om hade vi även övervägt att skriva PC-programmet i C istället för Java. Trots att C är ett nytt språk för oss hade detta antagligen underlättat enormt då serial kommunikation inte är något som stöds från början i Java. För att få det att fungera var vi tvungen att använda externa bibliotek i Java och vilket medförde många problem och tog upp mycket tid.

Projektet har varit extremt lärorikt och från att inte förstå någonting alls i början av februari till att faktiskt ha en fungerade prototyp 4 månader senare var fantastiskt. Projektet knöt samman kunskaper från tidigare kurser vilket gjorde att vi faktiskt tog tillvara på den kunskapen vi fått under utbildningen.