

Project 7 & 8
Tentamen registratie systeem

Paul Sohier 0806122
Sebastiaan Polderman 0820738

25 november 2010

Samenvatting

test

Inhoudsopgave

| | | |
|-------|-----------------------------|----------|
| 1 | Project omschrijving | 2 |
| 1.1 | <i>Software</i> | 2 |
| 1.1.1 | Computer | 2 |
| 1.1.2 | Microcontroller | 3 |
| 1.2 | <i>Communicatie</i> | 3 |
| 2 | Project verdeling | 3 |
| 3 | Project status | 4 |

1 Project omschrijving

Het doel van dit project en dit document is het uitleggen van de algemene status van het project en wat het project in het algemeen inhoudt. De opdracht van het project is het zelf verzinnen van een opdracht welke past in de projectomschrijving. Hierbij was het belangrijker dat het project tastbaar was en dat er tevens ook gebruik gemaakt werd van software welke zelfgeschreven was.

Wij hebben gekozen om een tentamen registratie systeem te maken. Dit idee is gebaseerd op een bij het CBR in gebruik zijnde systeem voor het afnemen van theorie examens. Het doel van het systeem is dat de Hoge School, en andere scholen, gebruik kunnen maken van dit systeem en daardoor eenvoudig meerkeuze tentamens kunnen afnemen. De student krijgt direct na het tentamen de uitslag te zien van zijn tentamen, en de docent hoeft het resultaat enkel nog te verwerken in het cijfer registratie systeem. Het systeem bestaand in principe uit twee onderdelen. Dit zijn:

- De computer. Deze handelt alle data af welke binnen komt vanuit de lossen kastjes waarop de studenten hun antwoord ingeven. Tevens weergeeft de computer de vraag aan de student en laat de computer zien hoeveel tijd de student nog heeft.
- Het tentamen kastje. De student vult hierop zijn antwoord in door op de bijbehorende knop te drukken voor een bepaald antwoord. De student krijgt hierna feedback van het kastje over welk antwoord is opgegeven. Deze feedback wordt gedaan in de vorm van een led bij de knop van het ingevulde antwoord. Zolang de tijd nog niet verstreken is kan de student tevens ook nog zijn opgegeven antwoord wijzigen door op een andere knop te drukken. Zodra de tijd over is is deze wijziging echter niet meer mogelijk. De feedback wordt pas aan de student gegeven zodra de computer bevestigt heeft dat hij het antwoord ontvangen heeft. Dit kan dus in theorie enige tijd duren. Zie hiervoor ook het deel met betrekking tot de communicatie welke gebruikt wordt.

De computer regelt het gehele tentamen en ontvangt via bedrade communicatie de resultaten van de studenten. Deze resultaten worden hierna lokaal op de computer opgeslagen en aan het eind van het tentamen aan de student op de computer weergegeven. De tentamen kastjes bevatten enkel en alleen en aantal knoppen, er zit geen LCD display op het kastje. Om toch ervoor te zorgen dat de goede student bij het goede tentamen kastje zit moet er van te voren door de docent worden gecontroleerd of de correcte student op de correcte plaats zit.

Het tentamen kastje bestaat uit een microcontroller welke de communicatie en knopdetectie afhandelt. Voor de rest is dit kastje een in principe zeer dom kastje. De hardware in dit kastje bestaat puur uit de microcontroller en het communicatie IC.

1.1 Software

De software bestaat uit twee delen. Op de computer draait software geschreven in java welke het hele tentamen weergeeft en bijhoudt wie welk antwoord heeft ingevuld. Op de microcontroller wordt gebruik gemaakt van de taal C.

1.1.1 Computer

De software van de computer is onderverdeeld in 3 delen:

- Vragen weergave
Dit deel van de software zorgt ervoor dat de vragen voor de studenten worden weergegeven aan de studenten zodat deze de vraag kunnen beantwoorden. Tevens zorgt het ervoor dat de tijd die nog over is wordt weergegeven op het scherm. Zodra deze tijd over is zal er even worden gewacht om te zorgen dat de kastjes de communicatie af kunnen maken totdat alle kastjes geantwoord hebben.
- Communicatie
Dit deel van de software zorgt ervoor dat de communicatie met de kastjes goed verloopt. Deze communicatie gebeurt via een RS485 bus. Voordat het naar de computer gaat wordt dit omgezet naar een RS232 verbinding welke aangesloten wordt op de seriële port van de computer. De computer is de zogenaamde “master” van de RS485 bus. Dit betekent eigenlijk dat hij de baas is op de bus en bepaald wie er op welk moment mag zenden. Zonder een master zal de communicatie over de bus niet goed verlopen. Zodra de software een antwoord binnenkrijgt zal hij controleren of

de binnengekregen data geldig is en hierna zal hij de data doorsturen naar de antwoordverwerking. Wanneer er bij de communicatie een foutmelding is ontstaan zal de software aan het betreffende kastje vragen of hij de data opnieuw zou willen versturen.

Zodra een antwoord binnen is gekomen zal de software tevens een reply geven naar het tentamen kastje om door te geven dat het antwoord was gegeven. Het kastje kan na deze reply ook weergeven dat het antwoord is ontvangen en verwerkt is.

Wanneer de tijd op is zal de software laten weten via een broadcast melding dat de tijd op is. Wanneer een gebruiker op dat moment nog geen antwoord had gegeven zal het kastje dit versturen. Zodra het systeem van alle kastjes een antwoord heeft begint het aan een nieuwe vragen zal hij weer via een broadcast melding dit melden aan alle aangesloten kastjes. Hiermee weten de kastjes dat ze het vorige antwoord weg moeten halen en dat ze zo een nieuw antwoord kunnen versturen. De software zal pas de nieuwe vraag stellen wanneer alle kastjes geantwoord hebben dat ze er klaar voor zijn. Op deze manier weten we zeker dat alle kastjes ook echt klaar zijn met hun lokale taken en dingen en er geen communicatie problemen zijn opgetreden. Wanneer een kastje niet antwoord op de broadcast melding zal de computer een specifiek aan dat gericht kastje een vraag stellen met of hij klaar is. Dit duurt echter wel langer, doordat een aparte melding verstuurd moet worden.

- Antwoordverwerking

De antwoordverwerking zorgt er voor dat de antwoorden welke binnenkomen correct verwerkt worden. Hierna worden deze opgeslagen om aan het eind verwerkt te kunnen tot een cijfer. Doordat het systeem weet welke student (Via het studentnummer) op welk kastje zat, kan hij eenvoudig een studentnummer gebaseerde lijst maken met de cijfers.

1.1.2 Microcontroller

De software van de microcontroller is globaal verdeeld in twee delen. Deze delen hangen echter wel vrij zwaar van elkaar af. Het doel van de microcontroller is om zo dom mogelijk te blijven. Dit betekent dat hij zelf geen validatie doet van gegevens of antwoorden. Dit wordt geheel gedaan door de computer. Hierdoor konden we het kastje simpel houden en kosten het minder tijd om de microcontroller te programmeren. Het programmeren van de microcontroller kost in principe meer tijd als het programmeren in java, wat gebruikt wordt op de computer.

- Communicatie

De communicatie in de microcontroller is eigenlijk het belangrijkste taak van de microcontroller. Zodra de student op één van de knoppen drukt wordt dit verstuurd over de bus. Hierna wordt er gewacht, zoals hierboven ook al beschreven, totdat er een goedkeuring terugkomt. Zodra deze er is zal de microcontroller de led van die knop aanmoeten zetten zodat de student kan zien dat het antwoord verwerkt is. Verder loopt de communicatie zoals hierboven uitgelegt is.

- Interrupts Door gebruik te maken van interrupts kunnen we eenvoudig zien of er een knop is ingedrukt. Zodra er een knop wordt ingedrukt genereert de microcontroller een interrupt welke je af moet vangen. Hiermee kan je eenvoudig zien welke knop is ingedrukt en hierna bepalen wat je ermee moet doen. Hierna moet enkel de interrupt flag nog goed terug gezet worden zodat je weer een nieuwe interrupt kan ontvangen. Wanneer dit niet gebeurt zal de microcontroller niet nogmaals een interrupt sturen, en is het dus mogelijk dat je interrupts mist.

De software van de microcontroller en de computer staat in principe geheel los van elkaar. Ze communiceren met elkaar via een RS485 bus welke geheel niet gerelateerd is aan de gebruikte software. Voor de microcontroller wordt deze bus direct via de UART van de microcontroller aangeroepen en voor de software wordt deze aangeroepen als seriële port. Voor beide hadden we ook een compleet andere manier van communiceren kunnen kiezen zonder de gehele software aan te passen.

1.2 Communicatie

2 Project verdeling

Het project is verdeeld over 3 belangrijke onderdelen, te weten:

- Electronica

- Programmeren microcontroller
- Programmeren computersoftware

Voor de electronica werken we in principe samen aan het ontwerp en de uitvoering. Hiervoor hebben we gekozen doordat we allebei ervaring met electronica hebben en dit het makkelijkst samen kunnen maken. Door deze samenwerking kunnen snel de electronica ontwerpen en uitvoeren.

De beide overige onderdelen zijn verdeeld over ons beide. Sebastiaan zorgt primair voor het programmeren van de computer, terwijl Paul primair werkt aan de software voor de microcontroller. Beide hebben over deze verschillende onderdelen ervaring met de verschillende benodigde heden voor het programmeren.

3 Project status