Inhaltsverzeichnis

1 Einleitung

1.1 Problemstellung

1.2 Eiswasserspeicher

1.3 Lösung

2 Raspberry PI

Mit dem Raspberry PI bietet die Raspberry PI Foundation einen Einplatinencomputer bereit, der für Experimente und Entwicklung von Programmen geeignet ist. Sein günstiger Preis von weniger als 40 Dollar und die Größe einer Kreditkarte tragen zu seinem Erfolg bei. Weiterhin existieren verschiedene Varianten des Computers, welche sich hauptsächlich in der Rechenleistung unterscheiden. Ursprünglich sollte der Raspberry PI zum Experimentieren von Studenten verwendet werden, fand jedoch schnell Anklag in divseren anderen Bereichen außerhalb von Universitäten.
Für diese Arbeit wurde ein Raspberry PI 2 vorgeschrieben, da er eine kostengünstige Alternative darstellt, um hardwarenahe Software zu verwirklichen. Hierbei tragen besonders die General Purpose Input/Output Pins eine wichtige Rolle. Weiterhin verfügt der Raspberry PI über eine Ethernet Schnittstelle, welche zur Kommunikation dient. Da es sich um einen Einplatinencomputer und nicht um einen klassischen Mikrocontroller handelt, findet sich auf dem Raspberry PI ein Betriebssystems wieder, welches eine Erleichterung für die Softwareentwicklung darstellt. Hierzu zählt insbesondere die Umsetzung des TCPI/IP Stacks.

2.1 Betriebssystem

Die Betriebssysteme für den Raspberry PI sind zahlreich. Am häufigsten ist jedoch eine Linux-Distribution anzutreffen. In diesem Projekt kommt Raspbian, ein auf Debian angepasstes Betriebssystem. Die Installation von Raspbian findet mit dem NOOBS-Installer, ein von der Raspberry PI Foundation bereitgestellter Installationsassistent, statt. Das Betriebssystem vereinfacht die Softwareentwicklung dahingehend, dass es verschiedene Schnittstellen für den Zugriff auf die Hardware anbietet. Die für dieses Projekt am wichtigsten Schnittstellen sind einerseits die GPIO Pins und andererseits der Zugriff auf das Netzwerk.

2.2 GPIO

Von den insgesamt 40 Pins sind 26 GPIO Pins. Bei den anderen handelt es sich Pins für die Stromversorgung. Abbildung 2.1 zeigt die Belegung der Pins des Raspberry PI 2. Durch die GPIO Pins wird in diesem Projekt eine S0-Schnittstelle implementiert. Diese Schnittstelle sendet Impulse, welche einen Leistungswert repräsentieren. Als Beispiel könnten 1000 Impulse die Leistung von 1 kW darstellen.

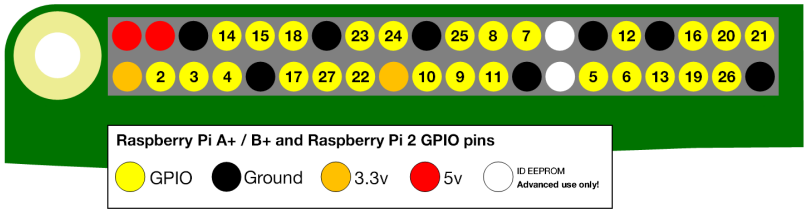

Abbildung 2.1: Pin Belegung Raspberry PI 2

Damit die GPIO Pins in einem C/C++ Programm angesprochen werden können, wird die Bibliothek *WiringPi* verwendet. Die Nummerierung der GPIO Pins unterschiedet sich zu der ursprünglichen Anordnung und ist aus Abbildung 2.2 ersichtlich.

P1: The Main GPIO connector							
WiringPi Pin	BCM GPIO	Name	Header		Name	BCM GPIO	WiringPi Pin
		3.3v	1	2	5v		
8	Rv1:0 - Rv2:2	SDA	3	4	5v		
9	Rv1:1 - Rv2:3	SCL	5	6	0v		
7	4	GPIO7	7	8	TxD	14	15
		0v	9	10	RxD	15	16
0	17	GPIO0	11	12	GPIO1	18	1
2	Rv1:21 - Rv2:27	GPIO2	13	14	0v		
3	22	GPIO3	15	16	GPIO4	23	4
		3.3v	17	18	GPIO5	24	5
12	10	MOSI	19	20	0v		
13	9	MISO	21	22	GPIO6	25	6
14	11	SCLK	23	24	CE0	8	10
		0v	25	26	CE1	7	11
WiringPi Pin	BCM GPIO	Name	Header		Name	BCM GPIO	WiringPi Pin

Abbildung 2.2: WiringPi Pinanordnung

Listing 2.1 zeigt ein einfaches Programm, welches *WiringPi* verwendet. Zeile vier zeigt eine Initialiserungsprozedur, welche einmalig aufgerufen werden muss, um auf die GPIO Pins zuzugreifen. Anschließend wird in Zeile fünf angegeben, ob auf dem Pin null lesend oder schreibend zugegriffen wird. Anschließend wird in einer Endlosschleife abwechselnd der Pin auf *HIGH* oder *LOW* gesetzt, jeweils mit einer Pause von 500 Millisekunden.

```
#include <wiringPi.h>
int main (void)
{
  wiringPiSetup () ;
  pinMode (0, OUTPUT) ;
  for (;;)
  {
    digitalWrite (0, HIGH) ; delay (500) ;
    digitalWrite (0,  LOW) ; delay (500) ;
  }
  return 0 ;
}
```

Listing 2.1: WiringPi Beispiel

2.3 Boost

Die C++ Bibliothek *boost* bietet diverse Funktionalitäten für die Entwicklung an. In diesem Projekt wird *boost* dazu verwendet, um auf das Netzwerk zuzugreifen und um Nebenläufigkeit zu ermöglichen.

3 Schaltung

Abbildung 3.1 zeigt den Schaltplan der Projekts. Auf der einen Seite ist der Raspberry PI erkennbar. Auf der anderen Seite befindet sich ein Arduino UNO, welcher einen Datenlogger darstellen soll. Da zur Zeit der Entwicklung der Datenlogger nicht vorhanden war, wurde mittels eines Arduino UNO dessen Funktionalitäten nachgebaut. Der Arduino UNO zählt die ankommenden Impulse und summiert diese auf. Per Konsole werden die Werte an den Entwickler gegeben, sodass er testen kann, ob die gewünschte Anzahl von Impulsen angekommen ist. Im Betrieb soll jedoch ein richtiger Datenlogger zum Einsatz kommen.
Der Kern des Schaltplans bildet der Optokoppler vom Typ KB Knighbright KB 817. Der Optokoppler trennt die beiden Stromkreise voneinander galvanisch. Diese Trennung sicheret den Raspberry PI gegen hohen Spannungen ab und verhindert somit seine Zerstörung.

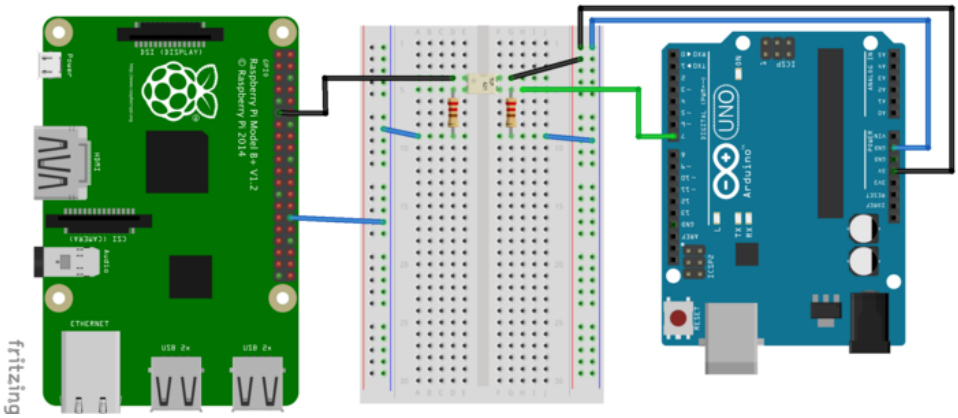

Abbildung 3.1: Schaltung

4 Software

4.1 Modell

4.2 Simulator

4.2.1 Eiswasserspeicher

4.2.2 Steuer-Server

4.3 Steuer-Client

5 Zusammenfassung