1. **Waschmaschinensteuerung**

Eine Waschmaschinensteuerung soll programmiert werden.

Die Waschmaschine wird mit folgenden Anschlüssen gesteuert:

|  |  |  |
| --- | --- | --- |
| ***Anschluss*** | ***Funktion*** | ***verbinden mit*** |
| Motor | 1 schaltet den Trommelmotor ein | PC1 |
| MotorFast | 1 schaltet auf Schleudern | PC0 |
| Pump | 1 schaltet die Laugenpumpe ein (abpumpen) | PC2 |
| Heater | 1 schaltet die Wasserheizung ein | PC3 |
| WaterIn | 1 schaltet den Wasserzulauf ein | PC4 |

Die Waschmaschine hat folgende Melder:

|  |  |  |
| --- | --- | --- |
| ***Anschluss*** | ***Funktion*** | ***verbinden mit*** |
| Temp | Temperaturwahlschalter 0 bei 60°C, 1 bei 90°C | PC8 |
| Temp60 | Thermokontakt 1 wenn 60°C erreicht wurde | PC9 |
| Temp90 | Thermokontakt 1 wenn 90°C erreicht wurde | PC10 |
| W1 | Schwimmerkontakt meldet mit 1 wenn wenig Wasser in der Trommel ist | PC11 |
| W2 | Schwimmerkontakt meldet mit 1 wenn die Trommel voll Wasser ist | PC12 |
| OnOff | Meldet ob ein- oder ausgeschaltet ist. (Ein=1, Aus=0) | PC13 |

Zusätzlich Stromversorgung der Waschmaschine:

|  |  |  |
| --- | --- | --- |
| GND | Masse, Ground, 0V | GND |
| Vcc | 5V-Anschluss | 5V |

Bei den Steueranschlüssen muss der Mikrocontroller 0 oder 1 ausgeben, um etwas zu bewirken.

Die Melder werden von der Waschmaschine auf 0 oder 1 gesetzt. Daran kann der Mikrocontroller Zustände der Waschmaschine ablesen.

1. Programmablauf:

* 1. alles ausschalten
  2. Warten auf OnOff=1
  3. 8 Sekunden abpumpen (mov R0,#8000; bl warte8s)
  4. Pumpe aus, Motor an langsam, Wasserzulauf ein
  5. warten auf voll
  6. Wasserzulauf aus
  7. Heizung ein
  8. wenn Wahlschalter Temp = 1 dann 90°C sonst 60°C
  9. warte auf Temperatur erreicht
  10. Heater aus
  11. 8 Sekunden waschen
  12. Abpumpen
  13. Warten auf leer
  14. Schleudern (MotorFast=1)
  15. 8 Sekunden Warten
  16. Alles Aus
  17. Warten auf OnOff=0
  18. Beginn von vorne

2. Programmgerüst:

syntax unified

**.include** "../src/regs.asm"

**.global** main

//Namenszuweisungen mit .equ (equals) für besserer Lesbarkeit des Programms

//Ausgänge

.equ MotorFast,Bit0

//weitere Ausgänge

//Eingänge

.equ Temp,Bit8

//weitere Eingänge

**main:**

**schleife:**

bl LCD\_i2c\_clear //LCD-Display löschen

//alles ausschalten

mov R0,#0

ldr R1,=GPIOC

str R0,[R1,ODR]

ldr R0,=t1 //Text1 anzeigen

bl LCD\_i2c\_textzeile1

**Aus:**

**ldr R0,[R1,IDR]**

**tst R0,OnOff**

beq Aus //Warten auf Schalter onoff = 1

...

//Prpgrammablauf aus 1.

b schleife

//Anzeigetexte

**t1:**

**.asciz** "Einschalten"

**t2:**

**.asciz** "Abpumpen "

**t3:**

**.asciz** "Fuellen "

**t4:**

**.asciz** "Heizen "

**t5:**

**.asciz** "Waschen "

**t6:**

**.asciz** "Abpumpen "

**t7:**

**.asciz** "Schleudern "

**t8:**

**.asciz** "Fertig "

.end

Aufgaben:

1. Erstellen Sie den Programmablaufplan des Steuerungsprogramms (sehr wichtig).
2. Programmieren Sie in Assembler.

