

WSxx_PrjXX Tolle Projektidee

Student Group

| First Name | Surname | Matrikel Nr. |
|------------|---------|--------------|
| | | |
| | | |
| | | |

Table of Contents

| | |
|---|---|
| MEXLE2020 Modul : DummyName | 2 |
| Hardwareübersicht | 2 |
| <i>Neu erstellte Module</i> | 3 |
| Softwareübersicht | 3 |
| Blockbilder | 3 |
| Projektübersicht | 4 |
| Aufgabenstellung/Problemstellung | 5 |
| Pflichtenheft | 5 |
| Lösungsansatz | 5 |
| Erstellung der Hardware | 5 |
| Erstellung der Software | 5 |
| Überprüfung/Test | 6 |
| Offene Punkte | 6 |
| Resume | 6 |


MEXLE2020 Modul : DummyName



Dummy-Bild

Source: [Moyan Brenn @ Wikimedia.org](#)

(CC BY 2.0 Lizenz)



| | |
|-------------------------|---|
| Ersteller | ??? |
| letzte Änderung | Sommersemester 2019 |
| Kurzbeschreibung |  In der Kurzbeschreibung soll die wesentliche Funktion des Projekts in einem Satz zusammengefasst werden |



ACHTUNG! Für die Niederschrift bietet sich folgendes Vorgehen an:

1. erst Bilder, Diagramme und Tabellen erstellen und den Kapiteln zuordnen
2. dann mit dem Text beginnen!
3. Auch Fehler dürfen beschrieben werden
4. Es sollte soviel beschrieben werden, dass nachfolgende Studierende ohne Rückfragen Ihren Weg und Ziel nachvollziehen können.
5. Keine Ich-Form, Emotionen vermeiden!



6. Mit  ist Text gekennzeichnet, der in der finalen Doku nicht mehr vorhanden sein sollte
7. Mit  ist Text gekennzeichnet, der für die finale Doku ergänzt werden muss

Hardwareübersicht

Die Hardwareübersicht soll einen Überblick über die verwendete Elektronik (und ggf. mechanische Hardware) geben.



Neu erstellte Module



Delete!

Falls es von Ihnen neu erstellte Elektronik-Module gab, so sind diese hier kurz zu beschreiben. Für eine ausführliche Beschreibung wird eine neue Seite für jedes Modul angelegt.

Softwareübersicht



Delete!

Beschreiben Sie kurz, auf welchem Chip / Board die Software läuft. z.B.: In diesem Projekt wurde die Platine [mmc_1x1_328pb](#) verwendet und programmiert.

Blockbilder



Fix Me!

Modulblockbild

```
<uml> [*] -right-> main
```

```
state "void main(void)" as main {
```

```
state "Initialisierung" as main_Init
main_Init -right-> Programmschleife : Wenn kein Fehler, dann
state "Programmschleife" as Programmschleife {
  10ms : springe zu input
  100ms : springe zu calculate\nspringe zu was anderem
  10ms -down-> input
  100ms -down-> calculate
}
```

}

state "void input(void)" as input {

```
state "ADC initialisieren" as input1
state "irgendwas einlesen" as input2
input1 --> input2
input1 -left-> ADC_init
```

}

state "void ADC_init(void)" as ADC_init { }

state "void calculate(void)" as calculate { }

state "void init_calculate(void)" as init_calculate { } calculate-->init_calculate </uml> <uml>

state "ISR(TWI_vect)" as TWI_Interrupt {

```
state "I2C Status auslesen" as TWI1
state "Abhängig vom Status Dinge tun" as TWI2
TWI1 --> TWI2
```

}

</uml>

Tipps zum Chart-Plugin: <http://plantuml.com/de/>

Statemachine

```
<uml> [*] --> Init Init --> [*] State1 : this is a string State1 : this is another string
```

```
State1 --> State2 State2 --> [*] </uml>
```

Tipps zum Chart-Plugin: <http://plantuml.com/de/>

```
42. void main () {
43.   printf ("Hello World!");
44.   exit 0;
45. }
```

Erklärung zum Code:

- exit 0; kann weggelassen werden.

Projektübersicht

Aufgabenstellung/Problemstellung



Die Aufgabenstellung war eine kompakte Controllerplatine zu erstellen, welche als Basis für unterschiedlichste Projekte genutzt werden kann. Dabei sollte Wert auf Kompatibilität und größtmögliche Ausnutzung des verwendeten Microcontrollers werden.

Pflichtenheft



| Nr. | Projektziel | Beschreibung | Status | Priorität |
|-------|--------------------------------|---|---------------------|-------------|
| PH-01 | Motor wird angesteuert | Code implementiert, Elektronik funktionsfähig | erledigt | muss |
| PH-02 | Joystick wird eingelesen | Layout | offen | soll |
| PH-03 | Bewegungsablauf wird abgerufen | statt Joystick soll ein gespeicherter Bewegungsablauf auswählbar sein | erledigt | muss |
| PH-04 | Kamera erzeugt Bilder | Kamera auslösen | offen | kann |
| PH-05 | Speicherung der Bilder | Slot mit SD-Karte | nicht mehr relevant | ausgenommen |

Lösungsansatz



1. Wie wurde das Problem angegangen?
2. Wieso wurden die genutzten Bauteile und Platinen (z.B. MMC_328PB) verwendet?

Erstellung der Hardware



1. hier **nicht** Schaltplan / Board darstellen, sondern in den Unterseiten.
2. Auf was wurde bei der Systemauswahl Wert gelegt?
Welche Hürden wurden genommen?

Erstellung der Software



1. Auf was wurde Wert gelegt?
Welche Hürden wurden genommen?
2. hier sollte auch der Code beschrieben werden, welcher nicht explizit für die genutzte Hardware notwendig ist. Also zum Beispiel bei Spielen, die [Spielmechanik](#) oder bei einer Wetterstation die Aufteilung auf dem Display. In beiden Beispielen wäre die Beschreibung der Bibliothek der Anzeige nicht hier, sondern im Software-Kapitel der Anzeigehardware zu beschreiben.

Überprüfung/Test



1. Gab es Zwischenschritte / Zwischenergebnisse?
2. Wurden Testplatinen und/oder Testkonzepte erstellt?

Kurzanleitung zu WaveDrom:

<https://observablehq.com/@drom/wavedrom-ark?collection=@drom/wavedrom>

Offene Punkte



| Nr. | Bug/Feature | SW/HW | Beschreibung |
|---------------|-------------|-------|--|
| Iss-01 | Bug | SW | Bei der Eingabe eines 18-stelligen Codes werden die letzten 17 Stellen abgeschnitten |
| Iss-02 | Bug | HW | LED X3 funktioniert nicht -> Kurzschluss durch Lötzinn |
| Iss-03 | Feature | SW | Statemachine ist einzubauen |
| Iss-04 | Feature | HW | Kameraauslöser ist zu entwickeln (noch nicht verfügbar) |
| Iss-05 | Test | HW | SD-Karte noch nicht getestet |

Resumee



1. Fazit für das gesamte System
2. Verbesserungsvorschläge (Ausblick): Was könnte im Gesamtsystem noch umgesetzt werden?

From:

<https://wiki.mexle.org/> - **MEXLE Wiki**

Permanent link:

https://wiki.mexle.org/mexle/template_fuer_die_projektbeschreibung?rev=1569880587

Last update: **2021/05/09 09:57**

