

WSxx_PrjXX Tolle Projektidee

Student Group

First Name	Surname	Matrikel Nr.

Table of Contents

WSxx_PrjXX Tolle Projektidee

Hardwareübersicht

Neu erstellte Module

Softwareübersicht

Blockbilder

Projektübersicht

Aufgabenstellung/Problemstellung

Pflichtenheft

Lösungsansatz

Erstellung der Hardware

Erstellung der Software

Überprüfung/Test

Offene Punkte

Resume

2

3

3

3

3

5

5

5

5

5

6

6

6

WSxx_PrjXX Tolle Projektidee

 **Fix Me!**



Dummy-Bild

 **Fix Me!**

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

(CC BY 2.0 Lizenz)

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

 **Delete!**

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!

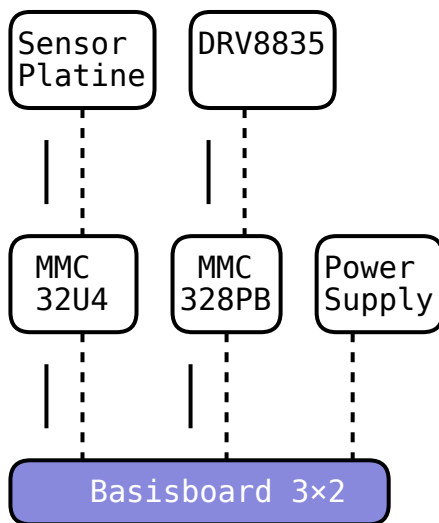


6. Mit  **Delete!** ist Text gekennzeichnet, der in der finalen Doku nicht mehr vorhanden sein sollte

7. Mit  **Fix Me!** 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



Bitte verfolgend sie die in Ihrem finalen Stand offenen Punkte (Bugs, fehlende Features, nicht vorgenommene Tests) in der [jeweiligen Seite der einzelnen Platine](#). Im Ausnahmefall (z.B. Architektur-Fehler wie fehlende Platinen) kann auch hier die Liste aus der Platine kopiert und gefüllt werden

Resumee



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

From:

<https://first.mexle.te.hs-heilbronn.de/> - MEXLE Wiki

Permanent link:

https://first.mexle.te.hs-heilbronn.de/mexle/template_fuer_die_projektbeschreibung?rev=1569881079

Last update: 2021/05/09 09:57

