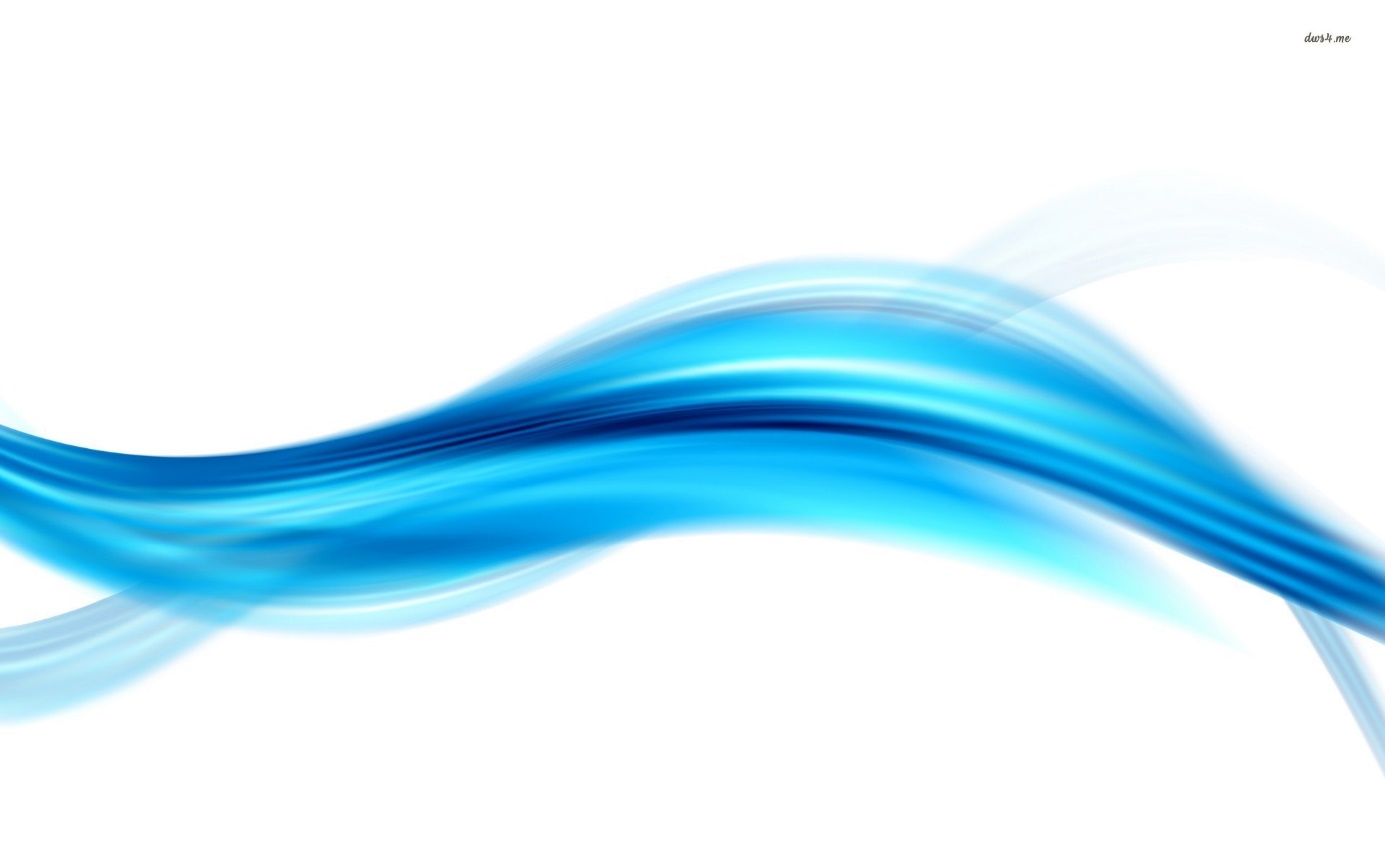
WAVE-O-MAT 2000

Team: Studiengang Systemtechnik

Oliver Schmid

Joёl Koch



# AUFGABENSTELLUNG

## RAHMENBEDINGUNGEN

Dies ist eine Abschlussarbeit in Embedded System, welche mithilfe des Roboters „miniQ 2WD“ gelöst werden soll. Der Roboter ist von DFrobot und ist im freien Markt erhältlich. Das System basiert auf einer Arduino Leonardo Plattform. Die Aufgabe besteht darin zwei Programme mit freiwählbaren Funktionen zu Programmieren. Die Ganzen Rahmenbedingungen sind im Dokument <<PA\_Aufgabenstellung\_mit\_Notizen>> zu entnehmen.

## IDEE

Der MiniQ verfügt über zwei Helligkeitssensoren, die als Spannungsteiler angeordnet sind. Dadurch kann über den analogen Eingang das Verhältnis der beiden Helligkeiten zueinander erfasst werden. Wird nun Beispielsweise. ein Finger gleichmässig hin- und her bewegt vor den Sensoren, können Grundfrequenz und Amplitude der Bewegung ermittelt werden.

Während das Programm 1 läuft, leuchten die RGB-LEDs. Die Helligkeit pulsiert dabei mit Frequenz, die aus der Bewegung ermittelt wurde.

Während das Programm 2 läuft, wird die Frequenz der Bewegung in einen hörbaren Bereich übersetzt und auf dem Lautsprecher als Ton ausgegeben. Die Eingangsfrequenz wird auf ein Band von 1 Hz bis 10 Hz limitiert und anschliessend auf einen Frequenzbereich von 100 Hz bis 10 kHz übersetzt.

## FUNKTIONSBESCHREIBUNG

### Funktionale Spezifikation für das Gesamtsystem WAVE-O-MAT

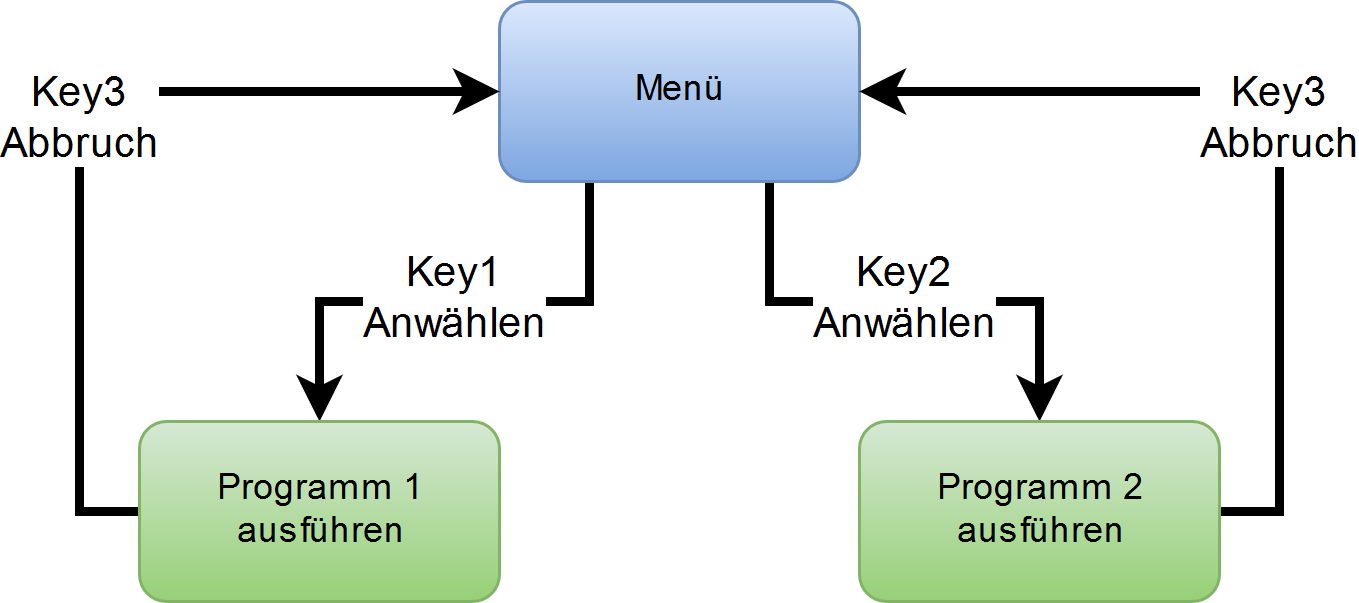
# Architektur

### MODULARISIERUNG

Alle Module Beschreiben

### Hauptprogramm

### Menügeführte Schnittstelle



### Tastenauswertung

Die Tastenauswertung ist ein eigenständiger Block in «Keypoll». Die miniQ-Tasten liegen an einem

Analog-Eingang an und erzeugen je nach Taste andere Spannungen. Diese werden durch die

vorgefertigte Funktion getKey() ausgelesen.

### LCD Anzeige

Die LCD Anzeige wird durch die Arduino-Bibliothek «LiquidCrystal\_I2C» angesteuert. Das Programm steuert den Cursor und die Ausgabe ans LCD. Ein Update des vollständigen Anzeigebereichs (32 Zeichen) dauert länger als 2 ms. Um dieses Problem zu umgehen, wird das Display Zeichen für Zeichen (resp. Cursorposition setzen) aktualisiert. Somit wird die 2ms Limite pro Updatevorgang nicht überschritten.

## MODULHIERARCHIE

## SCHNITTSTELLENDEFINITIONEN

Modul Flank

class Flank {

### public:

### Flank(double samplefrequency);

### double frequency(void);

### double frequencyraw(void);

### void calculate(int8\_t \* vdat);

### void printArray(int8\_t \* v, char \* title, uint8\_t len);

### void printTn();

### 

### private:

### void findFlanks(int8\_t \* vdat);

### void findExtrema(int8\_t \* vdat);

### void findMedian();

### void clearTn(void);

### void PrintVector(double \*vData, uint16\_t bufferSize, uint8\_t scaleType);Modul miniQdrive

void initDrive(void);

Modul Keypoll

int keypoll();

Modul pulselight

class Pulselight {

public:

void LED\_setup();

void LED\_end();

float LED\_setfq(float fq);

void LED\_update(float m);

Modul Keypoll

int keypoll();

Modul Keypoll

int keypoll();

### FUNKTIONALE SPEZIFIKATION

Die Funktionalen Spezifikationen dienen idealerweise als Test-Anweisungen für den Modul-, Integrations- und Abnahmetest.

# TESTSPEZIFIKATION GESAMTSYSTEM WAVE-O-MAT