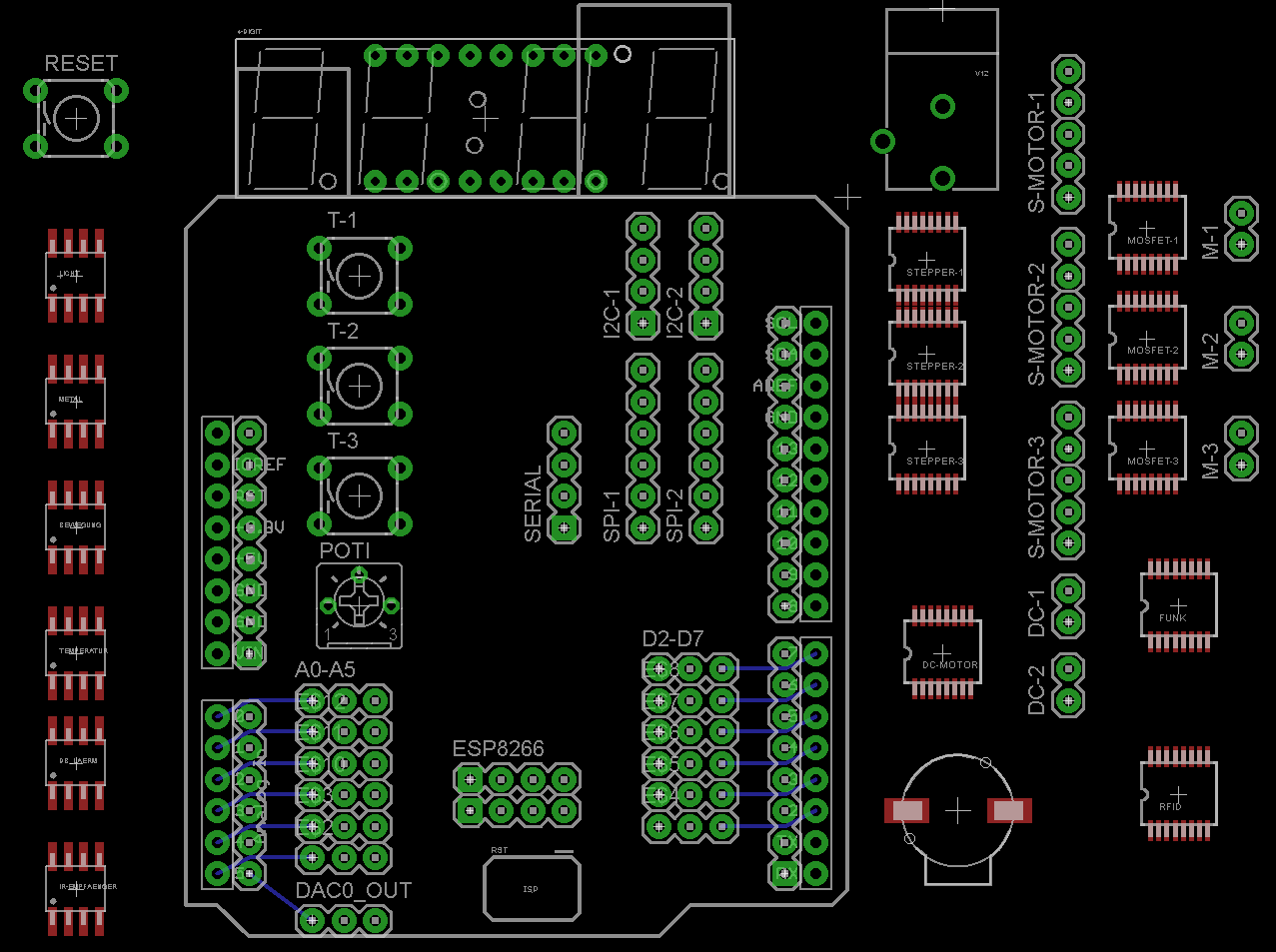
# IoTKit



## Allgemeines

* Das obenstehende Board ist als Skizze zu betrachten. Die Anordnungen und die verwendeten Komponenten dienen nur zur Visualisierung eines möglichen Aufbaus.
* Alle GND + und V+ der Sensoren und Aktoren sind verbunden
* Signale werden durch Steckerbrücken, bei Bedarf, verbunden
* Es ist ein Kit, d.h. Header müssen nicht zwingend eingelötet sein. Es ist auch zumutbar, dass diese vom Käufer nachträglich manuell eingelötet werden müssen.
* Das Board ist Arduino UNO V3 Header kompatibel. Es wird davon ausgegangen, dass alle Arduino Header 5-Volt fest sind.
* Zusätzliche Header für Freescale K64F sind vorhanden, aber nur teilweise verwendet.
* Die Verbindungsheader, um die Sensoren und Aktoren gegen die Signal zu schalten sind nicht eingezeichnet. Diese sollten als Connectors 2.54 mm ausgeführt werden.
* Die Einzelnen Sensoren und Aktoren sind zu Beschriften. Wünschenswert wäre auch eine Umrandung.
* Es ist ein Platz für ca. 3 Sponsoren Logo’s vorzusehen (darf auch unten sein).

**Links zu den Boards:**

* <http://developer.mbed.org/platforms/FRDM-K64F/> - Freescale FRDM-K64F
* <http://arduino.cc/en/Main/ArduinoBoardLeonardo> - Arduino Leonardo

Historie

Version 0.7

* MOSFET überarbeitet. LED Strip braucht Minuspol

Version 0.6

* Yahoo Wetterdaten holen

Version 0.5

* mbed RPC Beispiele, eingefügt und bestehende mit diesen gruppiert
* Raspberry Pi Kamera Beispiele.

Version 0.4

* Version mit Anwendungsbeispielen
* Problematisch Beispiel 2.2.7: evtl. weitere Pin Dreiergruppen an D8 und D9 vorsehen.

Version 0.3

* Erste schriftliche Version ohne Anwendungsbeispiele

## Beschreibung (von links oben nach rechts unten)

### Sensoren / Reset

|  |  |  |  |
| --- | --- | --- | --- |
| **Sensor** | **Typ** | **Pin** | **Beschreibung** |
| RESET | Switch | Reset | Reset Taster, Reboot Mikrokontroller |
| Licht | SC5549 oder GL5528 | A1 (K) | Lichtsensor – analog |
| Metall | SS49E | A2 (K) | Erkennen von magnetischen Feldern (HALL SENSOR MODULE) |
| ~~Bewegung~~ | ~~PIR Motion Sensor o.ä.~~ | ~~A3~~ | ~~Bewegungsmelder~~ |
| Temperatur | TMP75, LM77 | I2C (D14, D15) | Temperatursensor (Addr 0x48) |
| DB (Lärm) | ? | A3 (K) | Erkennen von akustischen Signalen, z.B. Klatschen, Hundegebell |
| Infrarot Empfänger | 38KHz Modulated | A5 | Infrarot Empfänger (Buchse) |
| Taster 1 | Switch | A0 | Taster 1 |
| Taster 2 | Switch | A1 | Taster 2 |
| Taster 3 | Switch | A2 | Taster 3 |
| Poti | Poti | A0 (K) | Poti |

### (K) = Kurzschlusstecker vorsehen

### Header

|  |  |  |  |
| --- | --- | --- | --- |
| **Header** | **Typ** | **Pin** | **Beschreibung** |
| Arduino UNO V3 | Header |  | Standard Arduino V3 Header (Leonardo Anordnung) |
| Freescale K64F | Header |  | <http://developer.mbed.org/platforms/FRDM-K64F/> - Header |
| A0-A5 | Connectors 2.54 mm | A0 - A5 | 6 x 3 Pins mit Signal, 5V, GND |
| DAC0\_OUT | Connectors 2.54 mm | DAC0\_OUT | Signal, V3.3, GND – Analoges Output Signal vom K64F |
| ESP8266 | Connectors 2.54 mm | A2, A3 | Ansicht von oben:    Folgende Pins sind verbunden:   * GND – GND * TX – A3 * GPIO 2 – NC * CH\_PD – V3.3 * RESET – NC * GPIO 0 – NC * VCC – V3.3 * RX – A2   ESP8266 soll so auf dem Kit gesteckt werden, dass er nach unten aus dem Kit herausschaut. |
| I2C-1 und IC-2 | Connectors 2.54 mm | D14, D15 | I2C-Bus  SCL – D15  SDA – D14  V – V3.3  GND – GND |
| Serial | Connectors 2.54 mm | D0, D1 | Serielle Schnittstelle  TX – D1  RX – D2  V – V3.3  GND – GND |
| SPI-1 | Connectors 2.54 mm | D13, D12, D11, D8 | SPI  SCK – D13  MISO – D12  MOSI – D11  SS – D8 |
| SP-2 | Connectors 2.54 mm | D13, D12, D11, D9 | SPI  SCK – D13  MISO – D12  MOSI – D11  SS – D9 |
| D2-D7 | Connectors 2.54 mm | D2 – D7 | 6 x 3 Pins mit Signal, 5V, GND |
| V12 | Power Jack |  | V12 Eingangssignal für MOSFET, Stepper- und Motordriver.  Umgewandelt als 5V für die entsprechend bezeichneten Pins und als Versorgungsspannung für dein Mikrokontroller |
| S-Motor 1 - 3 | Connectors 2.54 mm | Steppermotor Driver | 3 x 5-Pin Anschlüsse für unipolare Schrittmotoren. 4 x Treiberstufe und 1 x 5V (optimal: 5V oder 12V) |
| DC-1 und 2 | Connectors 2.54 mm | DC-Motor Driver | 2 x 2-Pin Anschlüsse für normale DC Motoren.  Ausgangssignal bis 12V bei 1 Ampere. OUT1+2 und OUT3+4 jeweils zusammen. |
| M-1 bis 3 | Connectors 2.54 mm | MOSFET | 3 x 2-Pin Anschlüsse für 12 Volt Verbraucher bis max. 1A. Signal und GND |

### Header (optional)

Wünschenswerte Header – falls einfach zu realisieren.

|  |  |  |  |
| --- | --- | --- | --- |
| **Header** | **Typ** | **Pin** | **Beschreibung** |
| ISCP | Header | D11-D13 | ISCP Header vom Leonardo, verbunden mit SPI Bus auf D11 – D13. Könnte verwendet werden um Arduino Ethernet Shield anzusprechen  (Fixe Position beachten). |
| RF / WiFi Expansion Header | Header |  | RF Expansion Pins vom K64F 1:1 nach oben führen  (genügend Platz für gesteckten ESP8266 berücksichtigen). |

### Aktoren

|  |  |  |  |
| --- | --- | --- | --- |
| **Aktor** | **Typ** | **Pin** | **Beschreibung** |
| 4-Digit Anzeige | SAA1064T  7-segment LED Common Anode | I2C | 4-Digit Anzeige  I2C (Addr 0x70) |
| Stepper 1 Motor Driver | ULN2003APG | D5, D4, D3, D2 | Schrittmotor Ansteuerung 1 (3 Pins sind unbenutzt). |
| Stepper 2 Motor Driver | ULN2003APG | PTC8, PTC1, PTB19, PTB18 | Schrittmotor Ansteuerung 2 (3 Pins sind unbenutzt). |
| Stepper 3 Motor Driver | ULN2003APG | PTC5, PTC7, PTC0, PTC9 | Schrittmotor Ansteuerung 3 (3 Pins sind unbenutzt). |
| DC Motor | L298N, L293B, SN754410 | D2 – D7 | DC Motor Ansteuerung (H-Brücke)   * EN1 – D3 (PMW) * IN1 – D2 * IN2 – D4 * EN2 – D5 (PWM) * IN3 – D6 * IN4 – D7   Andere Anordnung ist gewollt, weil D6/D7 dem gleichen Port zugeordnet sind. D2/D4 haben kein PWM Signal, D3 schon. |
| Buzzer | Piezo | D7 (K) | Buzzer |
| MOSFET 1 | TIP120 | D5 | MOSFET 1 - V12, 1A |
| MOSFET 2 | TIP120 | D6 | MOSFET 2 - V12, 1A |
| MOSFET 3 | TIP120 | D7 | MOSFET 3 - V12, 1A |
| Funk | SRF433/868 | D9 | Funk |
| RFID-Reader | NFC MFRC522 | D8, D10 – D13 | RFID-Reader   * SPI\_MOSI – D11 * SPI\_MISO – D12 * SPI\_SCLK – D13 * SPI\_CS – D10 * MF\_RESET – D6 |
| 4 x LED | SMD Led | D10 – D13 (K) | 4 verschieden Farbige LED (z.B. Rot, Gelb, Grün, Blau) |

## Anwendungen: Grundlagen

### Einfache I/O Anwendungen Digital

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Sensor/Aktor** | **Klasse** | **Pin’s** | **Beschreibung** |
| 1.1.1 | LED | DigitalOut | D13 | Blinkende LED (Grundaufbau Programm, Objekte und Klassen, *Ungarische* Notation)  Fortgeschrittene: Anwendung 1.5.1 |
| 1.1.2 | Taster  LED | DigitalIn  DigitalOut | A0  D13 | LED aufleuchten bei Tastendruck (if)  Hinweis: nicht vergleichbar mit Lichtschalter Zuhause, weil dieser i.d.R. Mechanisch ist.  Fortgeschrittene: Anwendung 1.5.2 |
| 1.1.3 | Taster  LED | DigitalIn  DigitalOut | A0  D13 | LED x Sekunden aufleuchten bei Tastendruck (if, wait)  Anwendung: Licht Treppenhaus. |
| 1.1.4 | Taster  Taster  LED | DigitalIn  DigitalIn  DigitalOut | A0  A1  D13 | LED mittels Tastendruck ein- und ausschalten. (if/else)  Anwendung: Lichtschalter |
| 1.1.5 | LED 1 – 4 | DigitalOut | D10 – D13 | Lauflicht (for, Rechnen)  Anwendung: Warnblinker, Wearables  Zusatzübung: mittels while implementieren  Fortgeschrittene: Anwendung 1.5.3 |
| 1.1.6 | LED 1 – 4 | DigitalOut | D10 – D13 | Melodie visualisieren (for Verschachtelt, wait)  Anwendung: Fernseher Simulation um Einbrecher abzuschrecken.  Fortgeschrittene: Anwendung 1.5.4 |
| 1.1.7 | Buzzer | DigitalOut | D7 | Melodie visualisieren (for Verschachtelt, wait)  Anwendung: Türglocke |

### Einfache I/O Anwendungen Analog

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Sensor/Aktor** | **Klasse** | **Pin’s** | **Beschreibung** |
| 1.2.1 | Poti  LED | AnalogIn  AnalogOut | A3  D13 | Helligkeit LED mittels Poti regeln (AnalogIn)  Anwendung: Dimmer |
| 1.2.2 | Taster  Taster  Poti  LED | DigitalIn  DigitalIn  AnalogIn  AnalogOut | A0  A1  A3  D13 | LED mittels Tastendruck ein- und ausschalten. (if/else, AnalogIn) mit Berücksichtigung Poti (PWM)  Anwendung: Lichtschalter mit Dimmer |
| 1.2.3 | Poti  LED 1 - 4 | AnalogIn  AnalogOut | A3  D10 – D13 | Lauflicht, Geschwindigkeit mittels Poti regeln.  Anwendung: Wearables , Baustellenwarnung |
| 1.2.4 | Poti  LED 1 – 4  Taster  Taster | AnalogIn  AnalogOut  DigitalIn  DigitalIn | A3  D10 – D13  A0  A1 | Lauflicht, Geschwindigkeit mittels Poti regeln. Ein-/Ausschalten mittels Taste  Anwendung: Baustellenwarnung |

### Serielle Schnittstelle (Ausgabe von Meldungen)

| **Nr.** | **Sensor/Aktor** | **Klasse** | **Pin’s** | **Beschreibung** |
| --- | --- | --- | --- | --- |
| 1.3.1 | USB | Serial | D0, D1 | Übungen mit Ausgabe von Meldungen auf der Seriellen Schnittstelle (printf). Z.B. ein Teil der vorherigen Anwendungen mit printf erweitern.  Anwendung: Debuggen |
| 1.3.2 | USB | Serial | D0, D1 | Statt Taster für vorherige Anwendungen ein Menu auf dem Terminal anzeigen und ‚e‘ für ein, ‚a‘ für aus und ‚-‚ und ‚+‘ für Dämmerungsschaltung verwenden.  Anwendung: Serielle Aktoren steuern, z.B. WLAN Modems (ESP8266) |
| 1.3.3 | USB | Serial | D0, D1 | Vorheriges Beispiel mit VT100 Positionierbefehlen. |
| 1.3.4 | USB | Serial | D0, D1 | Einfacher Rechner mit den Grundrechenoperationen. Die Eingabe erfolgt via Serielle Schnittstelle.  Bsp: 2 + 1<ENTER>, Ausgabe 3 |

### Sensoren, und Licht

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Sensor/Aktor** | **Klasse** | **Pin’s** | **Beschreibung** |
| 1.4.1 | Licht  LED 4 | AnalogIn  DigitalOut | A1  D13 | Licht bei Dämmerung einschalten  Anwendung: Dämmerungsschalter |
| 1.4.2 | Bewegung  LED 4 | DigitalIn  DigitalOut | A3  D13 | Wenn sich jemand nähert, Licht einschalten  Anwendung: Licht mit Bewegungsmelder, z.B. für Terassenbeleuchtung |
| 1.4.3 | Bewegung  LED 4  Buzzer | DigitalIn  DigitalOut  DigitalOut | A3  D13  D7 | Wenn sich jemand nähert, Licht einschalten und Akustisches Signal  Anwendung: Alarmanlage |
| 1.4.4 | Bewegung  LED 1 - 4  Buzzer | DigitalIn  DigitalOut  DigitalOut | A3  D10 – D13  D7 | Wenn sich jemand nähert, Lauflicht einschalten und Akustisches Signal  Anwendung: Alarmanlage |
| 1.4.5 | Bewegung  LED 1 + 2  Buzzer  Taster  Taster | DigitalIn  DigitalOut  DigitalOut  DigitalIn  DigitalIn | A3  D10 – D11  D7  A0  A1 | Wenn sich jemand nähert, Licht (D10) einschalten und Akustisches Signal.  Alarm ein-/ausschaltbar mittels Tastern. Wenn Alarm scharf ist, leuchtet LED 2  Anwendung: Alarmanlage |
| 1.4.6 | Metall  Buzzer | AnalogIn  DigitalOut | A2  D7 | Sobald kein magnetisches Feld mehr vorhanden, Akustisches Signal auslösen.  Anwendung: Fenstersicherung. |
| 1.4.7 | DB (Lärm)  LED 4 | AnalogIn  DigitalOut | A4  D13 | Bei akustischem Signal Licht ein-/ausschalten.  Anwendung: Klatschschalter |

### Programmlogik (nur Fortgeschrittene)

| **Nr.** | **Sensor/Aktor** | **Klasse** | **Pin’s** | **Beschreibung** |
| --- | --- | --- | --- | --- |
| 1.5.1 | LED | DigitalOut  Ticker | D13 | Blinkende LED (siehe 1.1.1) implementiert mittels Ticker. |
| 1.5.2 | Taster  LED | InterruptIn DigitalOut | A0  D13 | LED aufleuchten bei Tastendruck (siehe 1.1.2) implementiert mittels Interrupt |
| 1.5.3 | Poti  LED 1 – 4  Taster  Taster | AnalogIn  AnalogOut  DigitalIn  DigitalIn  Thread | A3  D10 – D13  A0  A1 | Lauflicht, Geschwindigkeit mittels Poti regeln. Ein-/Ausschalten mittels Taste (siehe 1.2.4).  Beim Einschalten wird ein Thread gestartet für das Lauflicht gestartet. Beim Ausschalten wird dieser gestoppt. |
| 1.5.4 | LED 1 – 4 | DigitalOut  struct {} | D10 – D13 | Melodie visualisieren mit struct und for.  Anwendung: Fernseher Simulation um Einbrecher abzuschrecken. |
| 1.5.5 | LED 1 - 4 | DigitalOut  Ticker  <*Object*> | D10 – D13 | 4 Ticker, LED gekapselt in einem Object mit einer Methode welche von Ticker aufgerufen wird. |

### Zeit RTC, LCD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Sensor/Aktor** | **Klasse** | **Pin’s** | **Beschreibung** |
| 1.6.1 | RTC |  |  | Setzen der Zeit und Ausgabe mittels printf  Anwendung: einfache Uhr |
| 1.6.2 | RTC, LCD | DigitDisplay | D8, D9 | Setzen der Zeit und Ausgabe Stunden und Minuten auf dem LCD |
| 1.6.3 | RTC, LCD | DigitDisplayTicker | D8, D9 | Wie 1.6.2 implementiert mittels Ticker |
| 1.6.4 | RTC, LCD, | DigitDisplayTicker  Serial | D8, D9 | Eingabe der Zeit mittels Serieller Console (Terminal) und Ausgabe auf LCD. |

### Sensoren, Internet, RTC, LCD und Licht

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Sensor/Aktor** | **Klasse** | **Pin’s** | **Beschreibung** |
| 1.7.1 | Ethernet  RTC | Ethernet-Interface NTPClient | - | Holen der Zeit vom Internet und interne RTC setzen. |
| 1.7.2 | Ethernet  RTC | Ethernet-Interface NTPClient  DigitDisplayTicker |  | Wie 1.6.3 aber vorher Zeit vom Internet holen.  Zusätzlich Zeit auf dem LCD ausgeben.  Anwendung: Uhr |
| 1.7.3 | Ethernet  RTC  LED 4 | Ethernet-Interface NTPClient  DigitDisplay  DigitalOut | D13 | Licht zeitgesteuert Ein- und Ausschalten, z.B. h:m:45 Ein, h:m:50 Aus. |
| 1.7.4 | Ethernet  RTC  LED 4  Licht | Ethernet-Interface NTPClient  DigitDisplay  DigitalOut  AnalogIn | D13  A1 | Licht zeitgesteuert oder bei Dunkelheit Ein- und Ausschalten |

### Fernsteuerung mittels Internet (RPC, Fernbedienung mittels Internet-Browser)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Sensor/Aktor** | **Klasse** | **Pin’s** | **Beschreibung** |
| 1.8.1 | Ethernet  DigitalOut | Ethernet  DigitalOut | D10 – D13 | <http://developer.mbed.org/users/feb11/code/HTTP-Server/> - Standard RPC HTTP Server ausprobieren mit LED 1 - LED 4 |
| 1.8.2 | Ethernet  DigitalOut | Ethernet  DigitalOut | D10 – D13 | LED 1 und LED2 mittels Browser http://<ip-addr>/l1/write 1 | 0 Ein-, Ausschalten |
| 1.8.3 | Ethernet  DigitalOut | Ethernet  DigitalOut | D10 – D13 | Einfache HTML Seite mit zwei Buttons zum Ein-/Ausschalten der LED‘s |

## Anwendungen: Aufbau 1 – Komplexe Anwendungen

### Servo, Infrarot Fernbedienung (Anwendung Robotic: Kamera Plattform)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Sensor/Aktor** | **Klasse** | **Pin’s** | **Beschreibung** |
| 2.1.1 | Servo 1 x  Taster 2 x | Servo  DigitalIn | A4  A0 + A1 | Servo Richtung, Tasten links, rechts. |
| 2.1.2 | Servo 1 x | Servo | A4 | Servo kalibrieren |
| 2.1.3 | Servo 2 x | Servo | A4 + A5 | Kamera Plattform bewegen |
| 2.1.4 | Servo 2 x  Infrarot FB | Servo  IR | A4, A5, PTB20 | Kamera Plattform mit IR Fernbedienung bewegen. |
| 2.1.5 | Ethernet |  |  | Anwendung 1.8.3 umbauen auf Kamera Plattform |

### DC Motor / (Anwendung: Robotic, Smart Robot Car)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Sensor/Aktor** | **Klasse** | **Pin’s** | **Beschreibung** |
| 2.2.1 |  | Motor | D2 – D4 | Motor 5 Sekunden vorwärts und rückwärts laufen lassen |
| 2.2.2 |  | Motor | D2 – D4 | Motor – Lauf(ruhe) kalibrieren |
| 2.2.3 |  | Motor  DigitalIn | D2 – D4  A0 – A2 | Motor mittels Tasten vor-, rückwärts und Stop |
| 2.2.4 |  | Motor  DigitalIn  AnalogIn | D2 – D4  A0 – A2  A3 | Motor mittels Tasten vor-, rückwärts, Stop und Poti Geschwindigkeit |
| 2.2.5 |  | Motor  DigitalIn  ultrasonic | D2 – D4  A0 – A2  A4, A5 | Wie 2.1.3, vor einem Hindernis Motor abstellen. |
| 2.2.6 |  | Motor  DigitalIn  ultrasonic | D2 – D4  A0 – A2  A4, A5 | Wie 2.1.6 aber langsam abbremsen. |
| 2.2.7 | Motor 2 x  SC04  Infrarot FB  Servo 2 x | Motor  Ultrasonic  IR  Servo | D2 – D7  A0 + A1  PTB20  A4 + A5 | Smart Robot Car und Kamera Plattform mittels Fernbedienung steuern. Droht der Car gegen eine Wand zu fahren, ist dieser zu stoppen. |
| 2.2.8 | Ethernet |  |  | Anwendung umbauen auf Smart Robot Card Anwendung. |

### Schrittmotoren (3D Drucker, Plotter, CNC-Fräsen etc.)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Sensor/Aktor** | **Klasse** | **Pin’s** | **Beschreibung** |
| 2.3.1 | Schrittmotor | Stepper-MotorUni | D2 – D5 | Schrittmotor jeweils um 90° Bewegen |
| 2.3.2 | Schrittmotor  Poti | Stepper-MotorUni  AnalogIn | D2 – D5  A3 | Schrittmotor jeweils um 90° Bewegen, Geschwindigkeit laut Poti setzen. |
| 2.3.3 | Schrittmotor  Microswitch | Stepper-MotorUni  DigitalIn | D2 – D5  A4 | Schrittmotor bis zum Endschalter laufen lassen, Nullpunkt setzen, 10 x 100 Schritte vorwärts und in einem Lauf zurück auf die Nullposition. |
| 2.3.4 | Schrittmotor  Microswitch | Stepper-MotorUni  DigitalIn  Serial | D2 – D5  A4  D0, D1 | Schrittmotor bis zum Endschalter laufen lassen. Befehle als einfachen G-Code von der Console (Terminal) scannen und ausführen. |

### Debugging und Trace

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Sensor/Aktor** | **Klasse** | **Pin’s** | **Beschreibung** |
| 2.4.1 |  |  |  | #define Debugging Beispiel |
| 2.4.2 |  |  |  | #define Trace Beispiel |

### Schalten grösserer Ströme, Fernbedienung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Sensor/Aktor** | **Klasse** | **Pin’s** | **Beschreibung** |
| 2.5.1 |  |  |  | Türschloss |
| 2.5.2 |  |  |  | RGB LED Strip 12V |
| 2.5.3 |  |  |  | Fernbedienung und RGB LED Strip 12V  Farben inkl. Zwischenfarben mittels Ziffern und Helligkeit mittels Up/Down einstellen. |
| 2.5.4 |  |  |  | DC Motor vor-, rückwärts und Servo links, rechts mittels Pfeiltasten steuern. |
| 2.5.5 |  |  |  | V220 Schaltsteckdose steuern |

### I2C

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Sensor/Aktor** | **Klasse** | **Pin’s** | **Beschreibung** |
| 2.6.1 |  |  |  | Temperator Sensor via I2C ansprechen |
| 2.6.2 |  |  |  | Kits mittels I2C verbinden |
| 2.6.3 |  |  |  | accelerometer and magnetometer (nur K64F). |

### SPI

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Sensor/Aktor** | **Klasse** | **Pin’s** | **Beschreibung** |
| 2.7.1 |  |  |  | NFC/RFID mittels SPI auslesen  Anwendung: Türschliessystem, Smart Cities (Reparaturmeldung an ERP). |
| 2.7.2 |  |  |  | Türöffner mittels RFID |
| 2.7.3 |  |  |  | LED Strip mittels SPI steuern |
| 2.7.4 |  |  |  | Sensordaten auf SD Karte speichern. |

### Fernsteuerung mittels Internet (RPC, Fernbedienung mittels Internet-Browser)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Sensor/Aktor** | **Klasse** | **Pin’s** | **Beschreibung** |
| 2.8.1 | Ethernet  DigitalOut | Ethernet  DigitalOut | D10 – D13 | <http://developer.mbed.org/users/feb11/code/HTTP-Server/> - Standard RPC HTTP Server ausprobieren mit LED 1 - LED 4 |
| 2.8.2 | Ethernet | Ethernet |  | Anzeige Temperatordaten im Browser |
| 2.8.3 | Ethernet | Ethernet |  | Fernbedienung mittels Internet (HTML5, jQueryMobile). |

## Anwendungen: Aufbau 2– «Dinge» und die Cloud

### Ethernet

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Sensor/Aktor** | **Klasse** | **Pin’s** | **Beschreibung** |
| 3.1.1 |  |  |  | IP-Adresse vom DNS Server holen |
| 3.1.2 |  |  |  | Fixe IP-Adresse zuweisen. |

### HTTP Protokoll

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Sensor/Aktor** | **Klasse** | **Pin’s** | **Beschreibung** |
| 3.2.1 |  |  |  | HTTP GET – Informationen holen |
| 3.2.2 |  |  |  | HTTP POST z.B. um Formulare abzuschicken |
| 3.2.3 |  |  |  | HTTP Server und Client. Sensordaten schicken und im 2. Kit auf SD Karte speichern. |
| 3.2.4 |  |  |  | Yahoo Wetterdaten holen ([http://weather.yahooapis.com/forecastrss?w=784794&u=c](http://www.google.com/url?q=http%3A%2F%2Fweather.yahooapis.com%2Fforecastrss%3Fw%3D784794%26u%3Dc&sa=D&sntz=1&usg=AFrqEzdvjNm7EQY83C7FDZ7cGLUZGF3qCw)) |

### REST

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Sensor/Aktor** | **Klasse** | **Pin’s** | **Beschreibung** |
| 3.3.1 |  |  |  | REST Aufruf zum lesen |
| 3.3.2 |  |  |  | REST Aufrufe zum schreiben |

### Mail

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Sensor/Aktor** | **Klasse** | **Pin’s** | **Beschreibung** |
| 3.4.1 |  |  |  | Alarm mittels Mail senden |
| 3.4.2 |  |  |  | Sensordaten mittels Mail versenden im CVS Format als Dateianhang. |

### Daten sammeln und auswerten

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Sensor/Aktor** | **Klasse** | **Pin’s** | **Beschreibung** |
| 3.5.1 |  |  |  | Google Spreedsheet via HTTP POST füllen |

### «Dinge» aus der Cloud steuern

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Sensor/Aktor** | **Klasse** | **Pin’s** | **Beschreibung** |
| 3.6.1 |  |  |  | HTML5, jQueryMobile Beispiel |
| 3.6.2 |  |  |  | Beispiel mit Cloud Anbieter XYZ |

### App‘s

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Sensor/Aktor** | **Klasse** | **Pin’s** | **Beschreibung** |
| 3.7.1 |  |  |  | HTML5, jQueryMobile Beispiel und Adobe HTML5 – nach APP Dienst |
| 3.7.2 |  |  |  | App von Cloud Anbieter XYZ |

## Anwendungen: Aufbau 3 – Raspberry Pi und Co. als Server

### Installation von Raspberry Pi und Co.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Sensor/Aktor** | **Klasse** | **Pin’s** | **Beschreibung** |
| 4.1.1 |  |  |  | Installation von SD Card |
| 4.1.2 |  |  |  | Feintuning, Sprache, Disk, Takt |
| 4.1.3 |  |  |  | ssh Server aktivieren |
| 4.1.4 |  |  |  | Feintuning – Software, (x11vnc, rdesktop) |

### NAS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Sensor/Aktor** | **Klasse** | **Pin’s** | **Beschreibung** |
| ~~4.2.1~~ |  |  |  | ~~Installation SAMBA~~ |

### Server Software und Smartphone Apps - FHEM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Sensor/Aktor** | **Klasse** | **Pin’s** | **Beschreibung** |
| 4.3.1 | USB |  |  | FHEM |
| 4.3.2 | USB |  |  | Board mit Raspberry Pi via USB verbinden |
| 4.3.3 | USB |  |  | FHE  M App |

### Server Software und Smartphone Apps – Eclipse Smarthome

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Sensor/Aktor** | **Klasse** | **Pin’s** | **Beschreibung** |
| 4.4.1 | Ethernet |  |  | Eclipse Smarthome / openHAB |
|  |  |  |  | Board via HTTP verbinden |
|  |  |  |  | Board via I2C verbinden |
| 4.4.2 | Ethernet |  |  | openHAB App |

### Raspberry Pi im Internet verfügbar machen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Sensor/Aktor** | **Klasse** | **Pin’s** | **Beschreibung** |
| 4.5.1 |  |  |  | Installation Apache Server |
|  |  |  |  | Apache als Reverse Proxy |
|  |  |  |  | Port in Router öffnen |

### Server Software MQTT und CoAP

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Sensor/Aktor** | **Klasse** | **Pin’s** | **Beschreibung** |
| 4.4.3 | Ethernet |  |  | MQTT und Moquitto als Ergänzung |
| 4.4.3 | Ethernet |  |  | CoAP und mbedServer als Alternative |

### Kamera anschliessen (USB, Raspberry Pi - Überwachung)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Sensor/Aktor** | **Klasse** | **Pin’s** | **Beschreibung** |
| 4.6.1 |  |  |  | Raspberry Pi Kamera – Anschliessen, Funktionen, Grenzen |
| 4.6.2 |  |  |  | USB Kamera |
| 4.6.3 |  |  |  | Kamera Standbilder erstellen |
| 4.6.4 |  |  |  | Kamera Streamen |

### SW Verteilung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Sensor/Aktor** | **Klasse** | **Pin’s** | **Beschreibung** |
| 4.7.1 |  |  |  | Programm via REST compilieren  http://developer.mbed.org/handbook/Compile-API |
| 4.7.2 | Ethernet |  |  | Apache als SW Release Server |