

Beschreibung MITASSIT System

1. Hardware

Die Hardware besteht aus dem Controllerboard mit WiFi und SD-Karte.

Hier wird die Stromversorgung und die Sensoren angeschlossen (Pinbelegung steht auf der Rückseite)
Alle Sensoren 8Motion + HF) werden parallel als Bus an die 4 Kontakte angeschlossen. Für EMG stehen weitere Kontakte zur Verfügung.

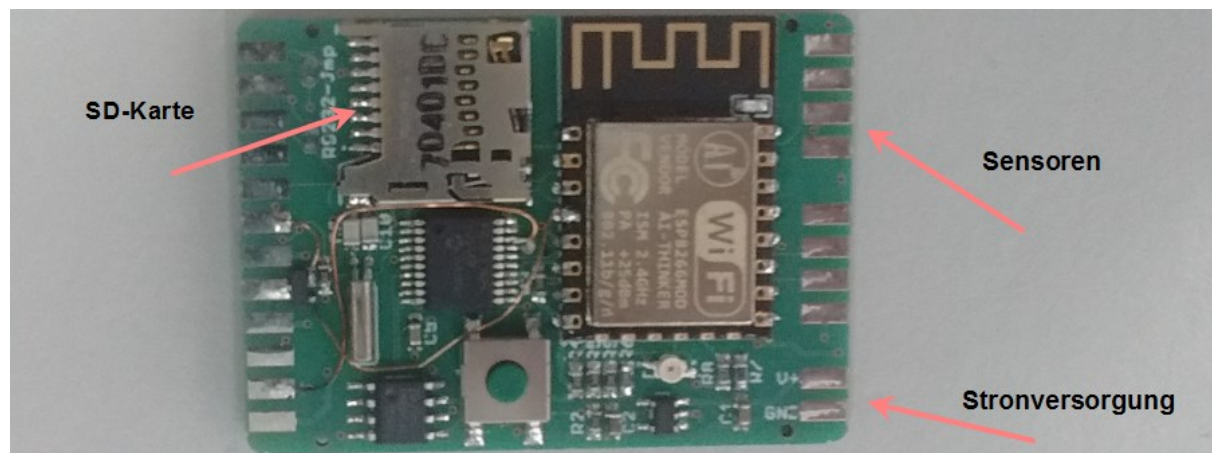
Die Sensoren werden an den gewünschten Stellen platziert und über dünne Drähte mit dem Controller verbunden.

Für EMG wird es später eine weitere kleine Platine geben, ebenso für den Feedback-Controller. Beide können an das controllerboard angeschlossen werden.

Die andere Seite des Controllerboards hat weitere Kontakte auf die ein Debug-Stecker zum Programmieren und debuggen angeschlossen werden kann. Für den Betrieb bleiben diese Kontakte frei.

Die Elektronik besteht aus einem Wifi-Modul (ESP), an den die SD-Karte angeschlossen ist. Daneben gibt es einen Sensor-Controller (PIC Microprozessor) welcher die Daten der Sensoren in Echtzeit abfragt und nach Zwischenspeicherung über die serielle Verbindung an den ESP zur Speicherung weitergibt. Das ESP-Modul wird über die Arduino Entwicklungsumgebung programmiert, der PIC über einen proprietären Compiler.

Beide Softwaremodule können bei Bedarf im Betrieb über das Webinterface upgedatet werden.



Die Stromversorgung kommt aus einem Wechselakku, es gibt einen Träger dafür, der, ebenso wie die Platine auf der Manschette festgenäht werden kann.



Der Akku kann eingesnapt und gewechselt werden, zum Laden steht ein Ladegerät zur Verfügung, so dass immer ein Akku geladen und der andere benutzt werden kann.

2. Konfiguration

Die Konfiguration des Systems wird auf einer Datei auf der SD-Karte abgelegt.
Für die Erstkonfiguration empfiehlt sich, diese Datei mit einem Texteditor anzupassen, insbesondere die WiFi Zugangsdaten sollten zunächst hier richtig eingestellt werden.
Die Datei heißt config.txt und befindet sich im Basisverzeichnis der SD-Karte:

HOT-SSID	ssid	SSID des WLAN Netzwerks
WLAN-PASS	xxxxx	Passwort zum verbinden
DEBUG-UDP	192.168.3.51	IP der UDP Gegenstelle
A-LIVESIG	2	Intervall für Lebenszeichen-Telegramm in Minuten
TIMESTAMP	0	zyklische aktualisierung des Timestamps in Minuten, 0 ist aus. Wird dann mit jeder neuen Datei aktualisiert
REMOTE-DEBUG	0	Wenn 1 werden Debug-Meldungen über den UDP-Kanal übertragen
SWITCH-WLAN-OFF	60	Timeout der Wlan-Verbindung in s, nach dieser Zeit bei Inaktivität wird die Verbindung abgeschaltet (Strom sparen)
FILE-INTERVAL	30	Gibt an wie viele Minuten in einer Datei geschrieben werden sollen, danach wird diese geschlossen und eine neue aufgemacht.
LOCAL-HOSTNAME	MITASSIST	
IP-STATIC	192.168.3.122	Statische IP des Moduls, braucht man um pwer Webinterface zugreifen zu können
V-BINARY	1	Auswahl: 0 = Hexdaten, 1 ist Binärdaten
Y-FILE-TIME	0	
0-DATA_RATE_MOTION	30	Datenrate für die Motion-Sensoren in ms (20 = 50Hz)
1-DATA_RATE_HF	33	Datenrate für den HF-Sensor in ms (30 = 33Hz)
2-DATA_RATE_EMG	0	Datenrate für den EMG-Sensor in ms (20 = 50Hz)
3-DATA_RATE_TRANSFER	1000	Transfer-Abstand zwischen den Blöcken zwischen Sensor-Controller und WLAN-Modul, typ 1000 = 1s

Die Einstellungen können, sobald das Modul am WLAN angemeldet ist, auch über das Webinterface eingestellt werden. (siehe Web-Interface Beschreibung)
Lässt sich das System nicht am WLAN anmelden, so kann während des bootens (Akku einlegen) durch gedrückt halten der Taste eine besondere Betriebsart eingestellt werden.
Es wird in diesem Fall vom Modul ein Hotspot erzeugt (Ssid = "ESP-BASIS", password = "mitassist"); der Webserver kann dann unter der IP 10.10.10.10 angesprochen werden.

3. Web-Interface

Solange das Wlan eingeschaltet ist, kann das Gerät über das Web-Interface auf seiner IP angesprochen werden. Hat sich das WLAN per timeout ausgeschaltet, so kann es durch drücken der Taste wieder aktiviert werden.

MITASSIST Web Interface

SensESP V0.50

Einstellungen

OTA Update

OTA PIC

System Reset

Dateien

Befehle

Neue Datei

Timestamp

Allgemeine Einstellungen

Netzwerk

Wlan SSID	<input type="text" value="eplant"/>
Wlan Password	<input type="password" value="....."/>
UDP-IP	<input type="text" value="192.168.3.51"/>
UDP-Debug (0 ist aus, 1 Log, 2 Data)	<input type="text" value="1"/>
UDP-Lebenszeichen Intervall (min - 0 ist aus)	<input type="text" value="2"/>
Zeitstempel Intervall (min, 0 ist bei jedem Dateiwechsel)	<input type="text" value="0"/>
Binärdaten (0 ist HEX)	<input type="text" value="1"/>
WLAN Timeout (in s - 0 ist immer an)	<input type="text" value="60"/>
Statische IP (0=DHCP)	<input type="text" value="192.168.3.122"/>
Hostname für DNS	<input type="text" value="MITASSIST"/>
	<input type="text"/>
Datei-Intervall (min)	<input type="text" value="2"/>
	<input type="text"/>
Datei-Zeitpunkt (min) innerhalb Datei-Intervall	<input type="text" value="0"/>

Hier lassen sich über Einstellungen die oben beschriebenen Parameter kontrollieren. Außerdem kann von hier ein Update der Software vorgenommen werden.

Ein einfacher Datei-Browser erlaubt das einsehen und hochladen von Dateien auf die SD-Karte.

Über die Befehle kann das Anlegen einer neuen Datei sowie ein neuer Timestamp interaktiv ausgelöst werden.

4. UDP-Kommunikation

Die Kommunikation zwischen Controller und der App des Smartphones erfolgt über einen UDP-Kanal.

In den Einstellungen muss die IP-Adresse der Gegenstelle angegeben werden, die Kommunikation erfolgt auf Port 2000

Vom Controller werden diese Telegramme gesendet:

S time id	Startmeldung nach dem Booten: es wird die aktuelle Unix-Time sowie die uid des Controllers übergeben
L vpp	Es wird zyklisch (einstellbar) ein Lebenszeichen gesendet, dafür schaltet das Modul das Wifi kurzzeitig ein und danach wieder aus. Es wird ein Wert für die Batteriespannung übergeben
T	Dieses Telegramm zeigt an, dass die Taste gedrückt wurde, das WLAN wird dazu aktiviert und wird entweder durch timeout oder udp-Kommando wieder ausgeschaltet.
P vpp	Battery-Low Telegramm mit Spannungswert. Die Batterie sollte ausgetauscht werden. Dieses Telegramm kommt anstelle des Lebenszeichens wenn Spannung zu niedrig

Wenn das WiFi eingeschaltet ist, wird zyklisch auf UDP-Telegramme geprüft. Es werden diese Telegramme verstanden

OK	Mit diesem Telegramm wird das WLAN sofort ausgeschaltet
S0 oder S1	damit kann das Sammeln der Daten aus/ oder eingeschaltet werden
Wmin	Damit kann der Wlan-Timeout auf min eingestellt werden
p2num	damit kann der Zähler Dateikennung auf einen definierten Wert eingestellt werden, typ wird p20 verwendet, so dass die Dateien dann wieder mit data000 beginnen.
N	Damit wird die aktuelle Datendatei geschlossen und eine neue begonnen.

Dieses Interface kann bei Bedarf weiter ausgebaut werden, z.B. um Befehle für die Feedback-Motoren zu übertragen.

Debugging

Über den UDP-Kanal kann über den Parameter UDP-Debug = 1 zusätzlich Debug-Information übertragen werden. Die Debugging-Zeilen beginnen mit einem # so dass sie von den Telegrammen oben unterschieden werden können.

Dies ist bei der Inbetriebnahme hilfreich, da man sofort sieht, wie das Gerät arbeitet und sich verbindet. Bei Bedarf kann ich ein Debugging-Tool für udp bereitstellen, mit dem man das protokollieren kann.

5. Datenstruktur

Die Daten können binär oder als HEX-Datensätze übertragen und in Dateien auf der SD-Karte abgelegt werden.

Die binäre Übertragung braucht weniger Platz, die HEX-Darstellung erlaubt eine einfachere Segmentierung und Überprüfung der Daten.

Die Daten sind in einzelnen Telegrammen zusammengefasst, welche jeweils zu Beginn eine Typ-Kennung tragen, Dann folgt ein Zeitstempel und danach die Daten.

Die Datenmenge ist für den jeweiligen Telegrammtyp definiert und konstant.

Bei der HEX-Darstellung werden die Telegramme mit einem CR-Zeichen abgeschlossen, so dass sie einfach zu segmentieren sind.

Bei der Binär-Übertragung folgend die Daten kontinuierlich nacheinander und müssen entsprechend der einzelnen Längen segmentiert werden.

Der Zeitstempel ist ein 16bit wert, der in ms interne Zeit zählt. Diese ist nicht sehr präzise, es ist davon auszugehen, dass sich über längere Zeiten Differenzen zu der absoluten UNIX-Zeit ergeben.

Deshalb wird zyklisch, mindestens mit jeder neuen Datei ein absoluter Zeitstempel in den Datenstrom eingefügt - hier wird dann der interne ms-Zähler jeweils wieder auf 0 gesetzt.

Es gibt diese Telegrammartentypen:

Typ	Hex-Kennung	Länge	
MPU_DATA_HEADER	0x55	27	
HF_DATA_HEADER	0xAA	69	
EMG_DATA_HEADER	0x33	53	
TS_DATA_HEADER	0x99	7	

Bei den MPU-Daten besteht das Telegramm aus einem Messpunkt mit 2 x 6 16-bit Werten = 24 byte, + Kennung + 2 byte Zeitstempel = 27 byte

Bei den anderen Sensordaten bestehen diese aus dem Header und einer definierten Anzahl von Messpunkten (33 bei HF und 25 bei EMG)

Beim Zeitstempel wird nach dem Header 4 byte Unix-Time + 2 Byte uid übertragen. Dadurch sind die Dateien eindeutig einem Sensor zuzuordnen.

Die Daten werden auf der SD-Karte in Dateien im Unterverzeichnis DATA geschrieben. Die Zeitdauer pro Datei ist einstellbar. Die Dateien data0000.txt werden fortlaufend durchnummeriert. Der Zähler steht im Eeprom, so dass die Dateien nicht überschrieben werden.

Wichtig: Wenn die Stromversorgung unterbrochen wird, so ist die letzte Datei nicht geschlossen und somit nicht lesbar. Vor dem Ende eines Versuches oder wenn aufgefordert wird, die Batterie zu tauschen, muss zuvor mit einem N Telegramm die aktuelle Aufzeichnung beendet werden!

Zunächst wird davon ausgegangen, dass die Dateien auf der SD-Karte verbleiben und durch Entnahme der SD-Karte auf einen PC übertragen werden.

In späteren Ausbaustufen muss geprüft werden, wie diese Daten auch über die WLAN-Verbindung zyklisch auf das Smartphone übertragen werden können - von der Bandbreite her ist das kein Problem.

6. Funktionsbeschreibung und Ablauf

Initialisierung

- Nach Power On geht die LED an.
- Bei SD-Karten-Fehler schnelles Blinken, System geht in Hotspot-Mode
- Es wird die Config-Datei gelesen und mit den Zugangsdaten versucht eine WIFI-Verbindung aufzubauen
- Sobald die Verbindung steht, geht die LED in einen langsamen Blinkmodus über.
- Es wird ein Zeitstempel vom ntp abgerufen
- Es wird ein UDP-Telegramm abgesetzt mit einer Identifikation und dem Zeitstempel
- Der Sensor-Controller wird parametrierung mit den Intervall-Daten aus der SD.
- Es wird eine neue Datendatei zum Schreiben aufgemacht, in welche die gesammelte Daten geschrieben werden
- es wird die LED ausgeschaltet und der Datensammel-Betrieb aufgenommen

Normalbetrieb

- PIC sammelt Daten mit der für die entsprechenden Sensoren eingestellten Datenrate.
- Daten werden in Blöcken an den ESP übertragen und dort auf die SD-Karte geschrieben
- In zyklischen Abständen wird die Datei geschlossen und eine neue aufgemacht.
- in definierten Zeitabständen verbindet sich der ESP mit dem WIFI und sendet ein udp-Telegramm als Lebenszeichen bzw. kann vom udp ein Kommando empfangen.
- in definierten Zeitabständen wird ein neuer Zeitstempel an den Sensor-Controller gesendet, der damit seine interne Zähler zurücksetzt und den Zeitstempel in den Datenstrom einfügt.
- wird während des Betriebs die Taste gedrückt, so wird die WiFi Verbindung aufgebaut und ein udp-Telegramm gesendet, Wifi läuft bis timeout

Besondere Betriebszustände:

- wird während des Bootens der Taster erkannt, so wird ein Hotspot erzeugt mit definierten Zugangsdaten, LED geht aus - keine Funktion außer Webserver
- nach dem Ablauf der eingestellten Zeit wird der WiFi deaktiviert und der Webserver abgeschaltet.
- zyklisch (mit Lebenszeichen) wird der Ladezustand des Akkus überwacht. sinkt er unter einen bestimmten wert wird eine UDP-Meldung Battery-Low erzeugt

7. SW-Update

Es gibt 2 SW-Komponenten: die SW im SensorController (OIC) und die SW im WiFi Modul (ESP)
 Die ESP-Software hat die Endung .bin, sie kann über die Funktion OTA-Update ausgewählt werden und direkt aktualisiert werden.
 Die Software des SensorControllers hat die Endung .hex und muss zunächst mit der Funktion "Dateien" ausgewählt werden (Durchsuchen) und kann dann mit "Upload" auf die SD-Karte geschrieben werden.
 Dann kann man mit dem Knopf "OTA PIC" die Software aktualisieren. Die Datei wird danach gelöscht.

8. Getting started

- define what WiDi Network shall be used, this may be a hotspot from a smartphone or the local net.
- you need the credentials of the net, the IP of the device and the IP of the udp recipient
- open config.txt file on the SD-card and edit it. Enter WiFi credentials and IPs.
- insert SD-card and start with Power-On.
- > in this phase you may listen at udp-port 2000 on the device that was defined in config.txt as udp partner when udp-debug is set to one.
- as soon as data is acquired the green LED at the HF-Sensor gets on. (if HF data acquisition is on)
- The key on the board wakes WiFi if it was switched off by timeout, so you may access udp and the web server.
- Before you switch off the power you should initiate a new data file, either by the web interface or a udp-telegram „N“. This closes the active data file. If not the last data file will not be readable.