



Berner Fachhochschule
Haute école spécialisée bernoise
Bern University of Applied Sciences

KlimaLogg Pro on Beagle Bone Black

Integrationsprojekt CAS EBX FS 2015 vom 30.09.2015
Christian Binder, Daniel Reimann, Urs Suhner

Aufgabenstellung

Eingabe für das Integrationsprojekt

Thema: Entwicklung eines Linux-Treibers für den "KlimaLogg Pro" Datenlogger:
<http://tfa-dostmann.de/index.php?id=81>

Aufbauend auf unserer Projektarbeit im Kurs "Embedded Linux" möchten wir den Phyton Treiber mit einem Linux Treiber ersetzen.

Die Userspace Applikation soll folgende Funktionalität auf dem BeagleBoneBlack zur Verfügung stellen:

- Speichern der Sensordaten in eine SQLite Datenbank
- Visualisieren der Sensordaten als Line-Graph in einer Qt-Applikation

Aufgaben Kernel-Treiber:

- Erkennen des USB-Transceivers
- Pairing des USB-Transceivers mit dem "KlimaLogg Pro" Datenlogger
- Datentransfer initiieren und Daten dem Userspace zur Verfügung stellen (Python Protokoll nachbilden)
- Die Userspace Applikation soll die Daten zyklisch vom Treiber lesen können



Kooperation/Synchronisation/Design

MASCOT Design

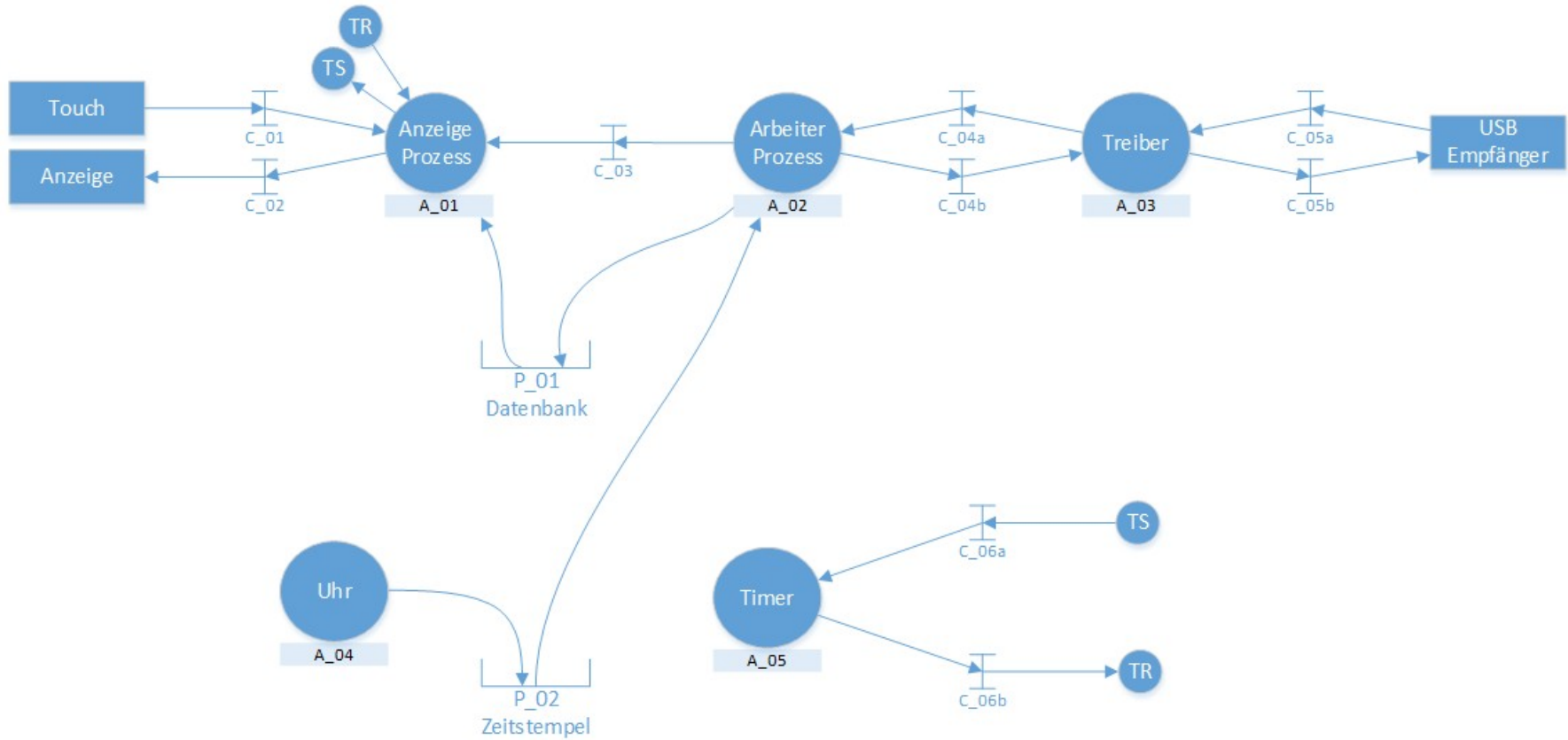


Abbildung 3: MASCOT Diagramm

Data Description

Index	MSB	LSB
10	LatestAddr	
11		
12		
13	thisAddr	
14		
15		
16	Pos_6_Hum_8 10	Pos_6_Hum_8 1
17	Pos_6_Hum_7 10	Pos_6_Hum_7 1
18	Pos_6_Hum_6 10	Pos_6_Hum_6 1
19	Pos_6_Hum_5 10	Pos_6_Hum_5 1
20	Pos_6_Hum_4 10	Pos_6_Hum_4 1
21	Pos_6_Hum_3 10	Pos_6_Hum_3 1
22	Pos_6_Hum_2 10	Pos_6_Hum_2 1
23	Pos_6_Hum_1 10	Pos_6_Hum_1 1
24	Pos_6_Hum_0 10	Pos_6_Hum_0 1
25	'\0'	Pos_6_Temp_8 10
26	Pos_6_Temp_8 1	Pos_6_Temp_8 0.1
27	Pos_6_Temp_7 10	Pos_6_Temp_7 1
28	Pos_6_Temp_7 0.1	Pos_6_Temp_6 10
29	Pos_6_Temp_6 1	Pos_6_Temp_6 0.1
30	Pos_6_Temp_5 10	Pos_6_Temp_5 1
31	Pos_6_Temp_5 0.1	Pos_6_Temp_4 10
32	Pos_6_Temp_4 1	Pos_6_Temp_4 0.1
33	Pos_6_Temp_3 10	Pos_6_Temp_3 1
34	Pos_6_Temp_3 0.1	Pos_6_Temp_2 10
35	Pos_6_Temp_2 1	Pos_6_Temp_2 0.1
36	Pos_6_Temp_1 10	Pos_6_Temp_1 1
37	Pos_6_Temp_1 0.1	Pos_6_Temp_0 10
38	Pos_6_Temp_0 1	Pos_6_Temp_0 0.1
39	PosDT_6_Year 10	PosDT_6_Year 1
40	PosDT_6_Month 10	PosDT_6_Month 1
41	PosDT_6_Day 10	PosDT_6_Day 1
42	PosDT_6_Hours 10	PosDT_6_Hours 1
43	PosDT_6_Minutes 10	PosDT_6_Minutes 1

Tabelle 5: History Record Beschreibung

Implementation Treiber

Ausgangspunkt: weewx Python Treiber

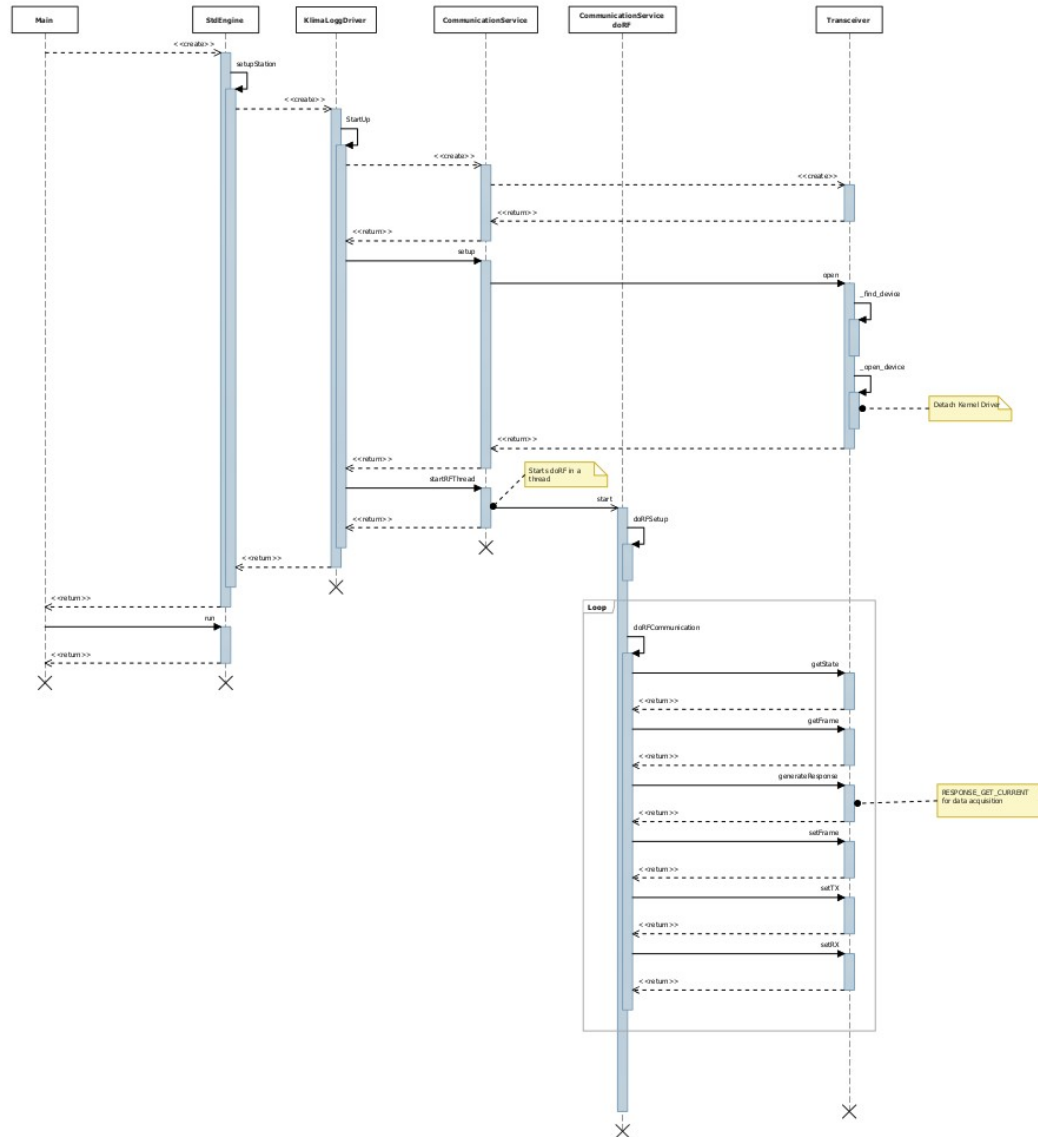
kl-106.py Treiber von Luc Heijst, Matthew Wall in der Version 1.0.6

Ein Teil der Funktionalität wurde im USB Treiber nachgebildet:

- init
- exit
- probe
- disconnect
- open
- read
- write
- release

Ein Teil wurde in der Qt Applikation (bitconverter.cpp) nachgebildet.

weewx Python Treiber Sequenz Diagramm



Wireshark USB PCAP

The image shows a Wireshark capture of USB traffic. The packet list on the left shows a 301-byte GET_REPORT response (packet 646777) selected. The packet details pane on the left shows the USB URB structure, including the USBPCap pseudoheader, IRP ID, USB status, URB function, and endpoint information. The packet bytes pane on the right shows the raw data in hexadecimal and ASCII.

Filter: Expression... Clear Apply Save Filter

No.	Time	Source	Destination	Protocol	Length	Info
646773	607.487867	host	3.0	USBHID	36	GET_REPORT Request
646774	607.487867	3.0	host	USBHID	34	GET_REPORT Response
646775	607.487867	3.0	host	USB	28	CLEAR FEATURE Status
646776	607.487867	host	3.0	USBHID	36	GET_REPORT Request
646777	607.487867	3.0	host	USBHID	301	GET_REPORT Response
646778	607.487867	3.0	host	USB	28	CLEAR FEATURE Status
646779	607.487867	host	3.0	USBHID	36	SET_REPORT Request
646780	607.487867	3.0	host	USBHID	49	SET_REPORT Response
646781	607.487867	3.0	host	USB	28	SET CONFIGURATION Status
646782	607.487867	host	3.0	USBHID	36	SET_REPORT Request
646783	607.487867	3.0	host	USBHID	301	SET_REPORT Response

► Frame 646777: 301 bytes on wire (2408 bits), 301 bytes captured (2408 bits)

▼ USB URB

USBPCap pseudoheader length: 28
IRP ID: 0xfffffa80081f2b80
IRP USBD_STATUS: USBD_STATUS_SUCCESS (0x00000000)
URB Function: URB_FUNCTION_CONTROL_TRANSFER (0x0008)
► IRP information: 0x01, Direction: PDO -> FDO
URB bus id: 1
Device address: 3
▼ Endpoint: 0x80, Direction: IN
1... = Direction: IN (1)
.000 0000 = Endpoint value: 0
URB transfer type: URB_CONTROL (0x02)
Packet Data Length: 273
Control transfer stage: Data (1)
[\[Request in: 646776\]](#)
[Time from request: 0.000000000 seconds]
[bInterfaceClass: HID (0x03)]

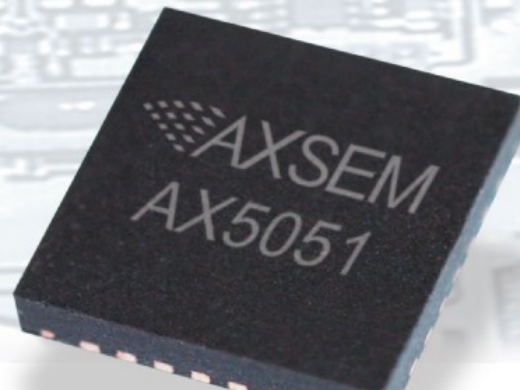
Offset	Hex	ASCII
0000	1c 00 80 2b 1f 08 80 fa ff ff 00 00 00 00 08 00	...+....
0010	01 01 00 03 00 80 02 11 01 00 00 01 00 00 b5 00
0020	42 01 40 64 1a 7b 10 ba e0 0b 8a 80 aa aa aa aa	B.@.{...
0030	aa 55 37 aa 57 0a aa aa aa aa aa aa aa 66 86 12	.U7.W...f..
0040	aa a6 57 15 08 29 03 42 aa aa aa aa aa aa 55 37 aa	.W..).B....U7.
0050	57 0a aa aa aa aa aa aa aa aa 66 86 12 aa a6 56 15	W.....f...V.
0060	08 29 03 41 aa aa aa aa aa aa aa 55 38 aa 57 0a aa aa	.).A....U8.W...
0070	aa aa aa aa aa aa 66 86 12 aa a6 56 15 08 29 03 40f...V..).@
0080	aa aa aa aa aa aa 55 38 aa 57 0a aa aa aa aa aa aaU8. W.....
0090	aa 66 86 12 aa a6 56 15 08 29 03 39 aa aa aa aa aa	.f...V. .).9....
00a0	aa 55 38 aa 57 0a aa aa aa aa aa aa aa aa 66 86 12	.U8.W...f...f..
00b0	aa a6 56 15 08 29 03 38 aa aa aa aa aa aa 55 38 aa	.V..).8....U8.
00c0	57 0a aa aa aa aa aa aa aa aa 66 96 12 aa a6 56 15	W.....f...V.
00d0	08 29 03 37 ac 1a aa aa aa 0a a4 aa 4a aa a4 aa	.).7....J...
00e0	4a aa aa aa aa aa aa 4a a4 aa aa 4a a4 aa aa aa	J.....J...J...
00f0	aa 0a a4 aa 4a aa a4 aa 4a aa aa aa aa aa 00 00	...J... J.....
0100	00 00 00 00 33 22 00 00 00 00 00 00 00 00 00 00	...3"...
0110	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0120	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

AXSEM AX5051 Datenblatt



Go farther, faster.

-116 dBm sensitivity @ 1.2 kbps
-103 dBm sensitivity @ 100 kbps
5 μ A RX current (duty cycled 1s, 100 kbps)



AX5051

400 - 470 MHz and
800 - 940 MHz universal, low
power, RF-Transceiver IC

USB 2.0 HID Specification / lsusb

sudo lsusb -v -d 6666:

Bus 003 Device 010: ID 6666:5555 Prototype product
Vendor ID

Device Descriptor:

bLength 18
bDescriptorType 1
bcdUSB 2.00
bDeviceClass 3 Human Interface Device
bDeviceSubClass 0 No Subclass
bDeviceProtocol 255
bMaxPacketSize0 64
idVendor 0x6666 Prototype product Vendor ID
idProduct 0x5555
bcdDevice 1.00
iManufacturer 1 LA CROSSE TECHNOLOGY
iProduct 2 Weather Direct Light Wireless Device
iSerial 3 0123456
bNumConfigurations 1
Configuration Descriptor:

...

Interface Descriptor:

...

bNumEndpoints 1
bInterfaceClass 3 Human Interface Device
bInterfaceSubClass 0 No Subclass
bInterfaceProtocol 0 None
iInterface 0

HID Device Descriptor:

bLength 9
bDescriptorType 33
bcdHID 1.11
bCountryCode 0 Not supported
bNumDescriptors 1
bDescriptorType 34 Report
wDescriptorLength 617
Report Descriptor: (length is 617)

...

Item(Global): Report ID, data= [0xd6] 214

...

Item(Global): Report Count, data= [0x10 0x01] 272

...

Endpoint Descriptor:

bLength 7
bDescriptorType 5
bEndpointAddress 0x81 EP 1 IN
bmAttributes 3
Transfer Type Interrupt
Synch Type None
Usage Type Data
wMaxPacketSize 0x0040 1x 64 bytes
bInterval 32

Device Status: 0x0100
(Bus Powered)

Implementation Userspace

SQLite 3.8.11.1 Datenbank

Kompilieren von SQLite fürs BBB

Download der Sourcen `sqlite-amalgamation-3081101.zip` von <https://www.sqlite.org/download.html> und entpacken der Files.
Mit angepassten `Makefile`, `make_env_host`, `make_env_target` Files aus dem Kurs Tools & Chains durchführen von:

```
make  
make install
```

Datenbankschema

```
CREATE TABLE measurement (  
  `dateTime` INTEGER NOT NULL UNIQUE PRIMARY KEY,  
  `temp0` REAL, ... `temp8` REAL, `humidity0` REAL, ... `humidity8` REAL);
```

```
CREATE TABLE parameter (`key` TEXT, `value` INTEGER);
```

QcustomPlot Qt C++ Widget (Graphen)

Einbinden von zwei Files ins Qt Projekt:

- qcustomplot.h
- qcustomplot.cpp

Demo/Abnahme

**ALLE SAGTEN:
DAS GEHT NICHT.
DANN KAM EINER, DER WUSSTE DAS
NICHT, UND
HAT'S EINFACH GEMACHT.**