Hochschule für Technik
und Wirtschaft Berlin
University of Applied Sciences

Testing and evaluation of communication technologies and protocols for microcontrollers in de-centralized, bottom-up DC microgrids.

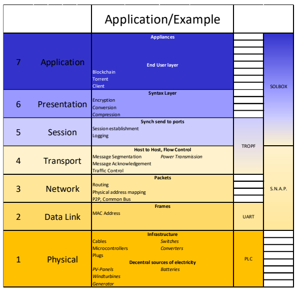
Referent, Datum
Seite 2

Content

- OSI model
- UART experiment
- PLC experiment
- PLC experiment DC/DC converter
- Decoding SIG60 PHY
- Fieldtest

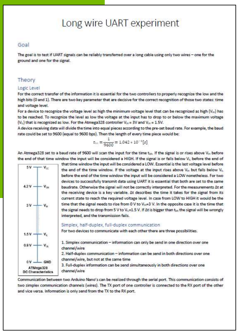
Referent, Datum
Seite 3

Die HTW Berlin
International



		Application/Example		
7	Application	**Appliances** **End User layer** Blockchain Torrent Client		SOLBOX
6	Presentation	**Syntax Layer** Encryption Conversion Compression		
5	Session	**Synch send to ports** Session establishment Logging	TROPF	
4	Transport	**Host to Host, Flow Control** Message Segmentation *Power Transmission* Message Acknowledgement Traffic Control		S.N.A.P.
3	Network	**Packets** Routing Physical address mapping P2P, Common Bus		
2	Data Link	**Frames** MAC Address	UART	
1	Physical	**Infrastructure** Cables *Switches* Microcontrollers *Converters* Plugs **Decentral sources of electricity** *PV-Panels* *Batteries* *Windturbines* *Generator*	PLC	

Referent, Datum
Seite 4



Long wire UART experiment

Goal

The goal is to test if UART signals can be reliably transferred over a long cable using only two wires – one for the ground and one for the signal.

Theory

Logic Level

For the correct transfer of the information it is essential for the two controllers to properly recognize the low and the high bits (0 and 1). There are two key parameter that are decisive for the correct recognition of those two states: time and voltage level.

For a device to recognize the voltage level as high the minimum voltage level that can be recognized as high (V_{IH}) has to be reached. To recognize the level as low the voltage at the input has to drop to or below the maximum voltage (V_{IL}) that is recognized as low. For the Atmega328 controller V_{IH} = 3V and V_{IH} = 1.5V.

A device receiving data will divide the time into equal pieces according to the pre-set baud rate. For example, the baud rate could be set to 9600 (equal to 9600 bps). Then the length of every time piece would be:

$$t_{bit} = \frac{1}{9600} = 1{,}042 \cdot 10^{-4}[s]$$

An Atmega328 set to a baud rate of 9600 will scan the input for the time t_{bit}. If the signal is or rises above V_{IH} before the end of that time window the input will be considered a HIGH. If the signal is or falls below V_{IL} before the end of that time window the input will be considered a LOW. Essential is the last voltage level before the end of the time window. If the voltage at the input rises above V_{IH} but falls below V_{IL} before the end of the time window the input will be considered a LOW nonetheless. For two devices to successfully transmit data using UART it is essential that both are set to the same baudrate. Otherwise the signal will not be correctly interpreted. For the measurements Δt at the receiving device is a key variable. Δt describes the time it takes for the signal from its current state to reach the required voltage level. In case from LOW to HIGH it would be the time that the signal needs to rise from 0 V to V_{IH}=3 V. In the opposite case it is the time that the signal needs to drop from 5 V to V_{IL}=1.5 V. If Δt is bigger than t_{bit} the signal will be wrongly interpreted, and the transmission fails.

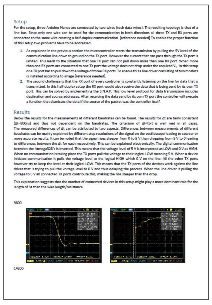
ATMega328
DC Characteristics

Simplex, half-duplex, full-duplex communication

For two devices to communicate with each other there are three possibilities.

1. Simplex communication – information can only be send in one direction over one channel/wire
2. Half-duplex communication – information can be send in both directions over one channel/wire, but not at the same time
3. Full-duplex information can be send simultaneously in both directions over one channel/wire

Communication between two Arduino Nano's can be realized through the serial port. This communication consists of two simplex communication channels (wires). The TX port of one controller is connected to the RX port of the other and vice versa. Information is only send from the TX to the RX port.



Setup

For the setup, three Arduino Nanos are connected by two wires (tech data wires). The resulting topology is that of a line bus. Since only one wire can be used for the communication in both directions all three TX and RX ports are connected to the same wire creating a half-duplex communication. [reference needed] To enable the proper function of this setup two problems have to be addressed.

1. As explained in the previous section the microcontroller starts the transmission by pulling the 5V level of the communication line down to ground on the TX port. However the current that can pass through the TX port is limited. This leads to the situation that one TX port can not pull down more than one RX port. When more than one RX ports are connected to one TX port the voltage does not drop under the required V_{IL}. In this setup one TX port has to pull down the voltage of three RX ports. To enable this a line driver consisting of two mosfets is installed according to image [reference needed].
2. The second challenge is that the RX port of every controller is constantly listening on the line for data that is transmitted. In this half-duplex setup the RX port would also receive the data that is being send by its own TX port. This can be solved by implementing the S.N.A.P. This low level protocol for data transmission includes destination and source addresses. After receiving the data send by its own TX port the controller will execute a function that dismisses the data if the source of the packet was the controller itself.

Results

Below the results for the measurements at different baudrates can be found. The results for Δt are fairly consistent (Δt≈800ns) and thus not dependent on the baudrates. The criterium of Δt<tbit is well met in all cases. The measured differences of Δt can be attributed to two aspects. Differences between measurements of different baudrates can be mainly explained by different step resolutions of the signal on the oscilloscope leading to coarser or more accurate results. It can be noted that the signal rises steeper from 0 to 5 V than dropping from 5 V to 0 leading to differences between the Δt for each respectively. This can be explained electronically. The digital communication between the Atmega328's is inverted. This means that the voltage level of 5 V is interpreted as LOW and 0 V as HIGH. When no communication is taking place the TX ports pull the voltage to their logical LOW meaning 5 V. When a device initiates communication it pulls the voltage level to the logical HIGH which 0 V on the line. All the other TX ports however try to keep the level at their logical LOW. This means that the TX ports of the devices work against the line driver that is trying to pull the voltage level to 0 V and thus delaying the process. When the line driver is pulling the voltage to 5 V all connected TX ports contribute this, making the rise steeper than the drop.

This explanation suggests that the number of connected devices in this setup might play a more dominant role for the length of Δt than the wire length/resistance.

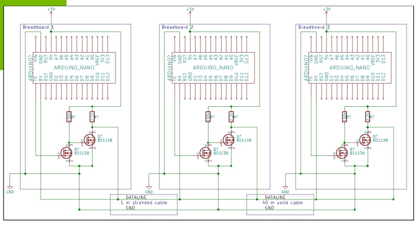
9600

14200

Referent, Datum
Seite 5

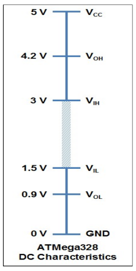
Long Wire UART Test

- Test of reliability of UART transfer over long cable
- Signal transfer inverted
- Simplex → Half-Duplex communication
- Line driver was used
- 1 meter stranded cable d=1.5 mm
- 40 meter solid cable d=1.5 mm



Referent, Datum
Seite 6

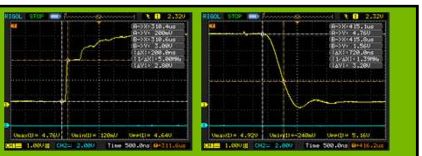
Logic Level



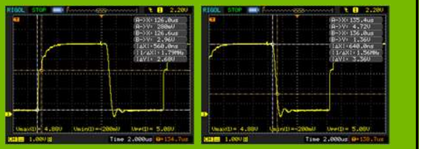
ATMega328
DC Characteristics

- Decisive are V_{IH} and V_{IL}
- Baudrate determines max Δt to rise above V_{IH} or drop below V_{IL}

htw
Referent, Datum
Seite 7



Baud rate 9600



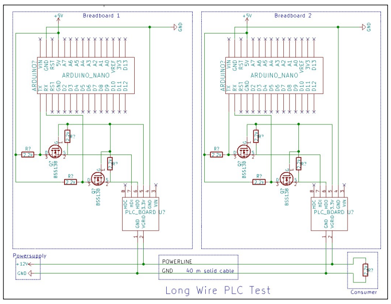
Baud rate 115200

---

htw
Referent, Datum
Seite 8

PLC

- Long Wire PLC Test

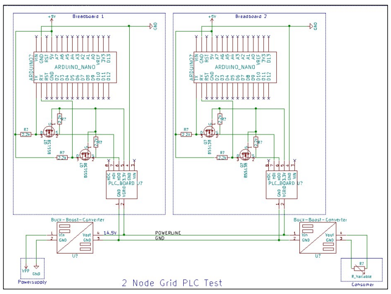

Breadboard 1
Breadboard 2
ARDUINO_NANO
PLC_BOARD
Powersupply
POWERLINE
GND
40 m solid cable
Long Wire PLC Test
Consumer

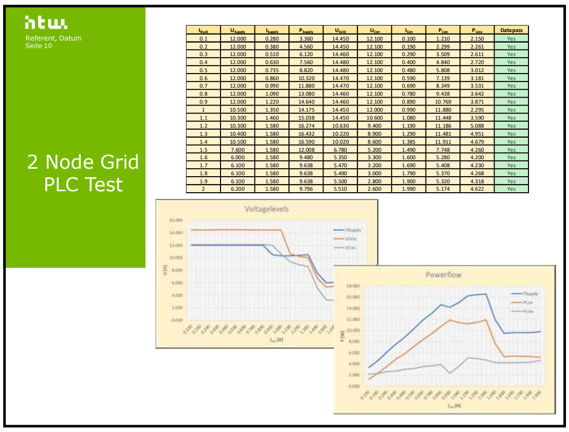
---

htw
Referent, Datum
Seite 9

2 Node Grid PLC Test

- 2 Node DC/DC converter test


Breadboard 1
Breadboard 2
ARDUINO_NANO
PLC_BOARD
Buck-Boost-Converter
POWERLINE
GND
Powersupply
R_Variable
Consumer
2 Node Grid PLC Test

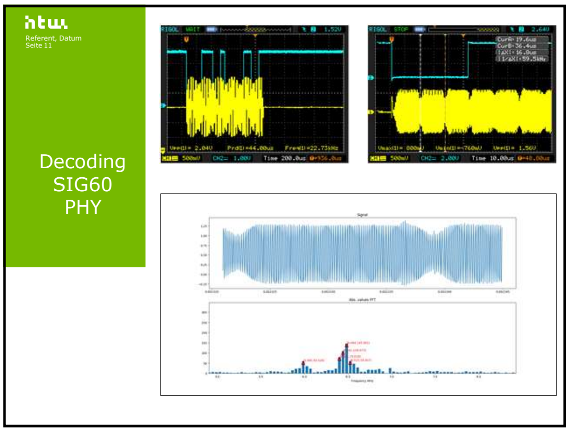


Referent, Datum
Seite 10

2 Node Grid PLC Test

I_{Buck}	U_{Supply}	I_{Supply}	P_{Supply}	U_{Grid}	U_{Con}	I_{Con}	P_{Con}	P_{Loss}	Datapass
0.1	12.000	0.280	3.360	14.450	12.100	0.100	1.210	2.150	Yes
0.2	12.000	0.380	4.560	14.450	12.100	0.190	2.299	2.261	Yes
0.3	12.000	0.510	6.120	14.460	12.100	0.290	3.509	2.611	Yes
0.4	12.000	0.630	7.560	14.480	12.100	0.400	4.840	2.720	Yes
0.5	12.000	0.735	8.820	14.480	12.100	0.480	5.808	3.012	Yes
0.6	12.000	0.860	10.320	14.470	12.100	0.590	7.139	3.181	Yes
0.7	12.000	0.990	11.880	14.470	12.100	0.690	8.349	3.531	Yes
0.8	12.000	1.090	13.080	14.460	12.100	0.780	9.438	3.642	Yes
0.9	12.000	1.220	14.640	14.460	12.100	0.890	10.769	3.871	Yes
1	10.500	1.350	14.175	14.450	12.000	0.990	11.880	2.295	Yes
1.1	10.300	1.460	15.038	14.450	10.600	1.080	11.448	3.590	Yes
1.2	10.300	1.580	16.274	10.630	9.400	1.190	11.186	5.088	Yes
1.3	10.400	1.580	16.432	10.220	8.900	1.290	11.481	4.951	Yes
1.4	10.500	1.580	16.590	10.020	8.600	1.385	11.911	4.679	Yes
1.5	7.600	1.580	12.008	6.780	5.200	1.490	7.748	4.260	Yes
1.6	6.000	1.580	9.480	5.350	3.300	1.600	5.280	4.200	Yes
1.7	6.100	1.580	9.638	5.470	3.200	1.690	5.408	4.230	Yes
1.8	6.100	1.580	9.638	5.490	3.000	1.790	5.370	4.268	Yes
1.9	6.100	1.580	9.638	5.500	2.800	1.900	5.320	4.318	Yes
2	6.200	1.580	9.796	5.510	2.600	1.990	5.174	4.622	Yes

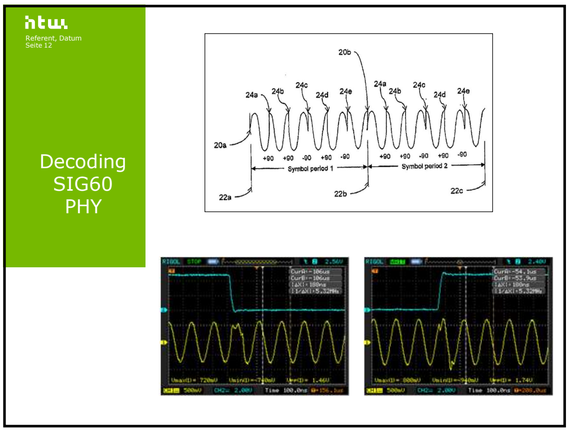
Voltagelevels
16.000
14.000
12.000
10.000
8.000
6.000
4.000
2.000
0.000
U[V]
USupply
UGrid
UCon
Powerflow
18.000
16.000
14.000
12.000
10.000
8.000
6.000
4.000
2.000
0.000
P[W]
PSupply
PCon
PLoss



Referent, Datum
Seite 11

Decoding SIG60 PHY

RIGOL
Signal



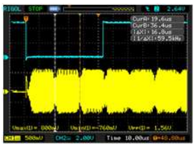
Referent, Datum
Seite 12

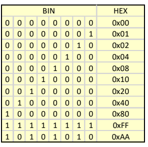
Decoding SIG60 PHY

20b
24a
24b
24c
24d
24e
24a
24b
24c
24d
24e
20a
+90
+90
-90
+90
-90
+90
+90
-90
+90
-90
Symbol period 1
Symbol period 2
22a
22b
22c
RIGOL
STOP
CurA: -106us
CurB: -106us
|ΔX|: 188ns
|1/ΔX|: 5.32MHz

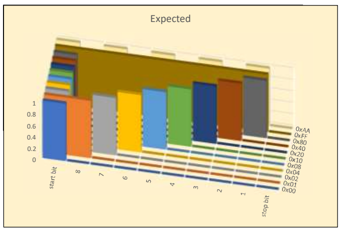
Referent, Datum
Seite 13

Decoding SIG60 PHY


RIGOL STOP
2.64V
CurA: 19.6us
CurB: 36.4us
|ΔX|: 16.8us
|1/ΔX|: 59.5kHz
Vmax(1) = 800mV
Vmin(1) = -760mV
Vpp(1) = 1.56V
CH1 500mV
CH2= 2.00V
Time 10.00us
+40.00us

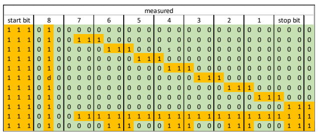


BIN								HEX
0	0	0	0	0	0	0	0	0x00
0	0	0	0	0	0	0	1	0x01
0	0	0	0	0	0	1	0	0x02
0	0	0	0	0	1	0	0	0x04
0	0	0	0	1	0	0	0	0x08
0	0	0	1	0	0	0	0	0x10
0	0	1	0	0	0	0	0	0x20
0	1	0	0	0	0	0	0	0x40
1	0	0	0	0	0	0	0	0x80
1	1	1	1	1	1	1	1	0xFF
1	0	1	0	1	0	1	0	0xAA

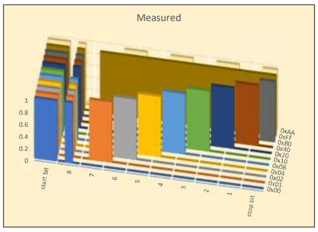


Referent, Datum
Seite 14

Decoding SIG60 PHY

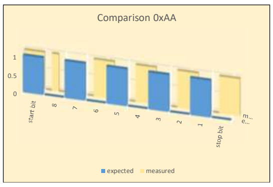


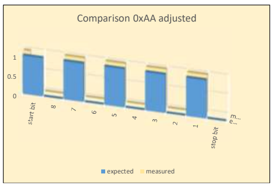
measured																														
start bit			8			7			6			5			4			3			2			1			stop bit			
1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	1	0	0	0	0	0	1	1	1	0	0	0	s	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	d	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0
1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
1	1	1	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	0	0	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1

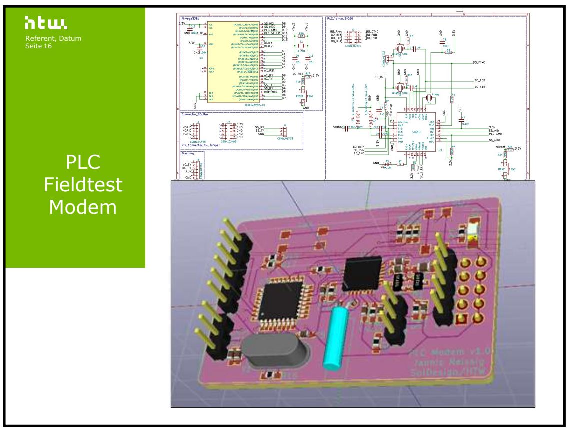


Referent, Datum
Seite 15

Decoding SIG60 PHY

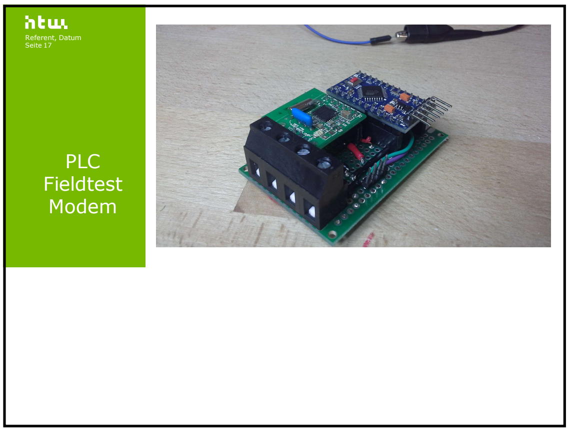






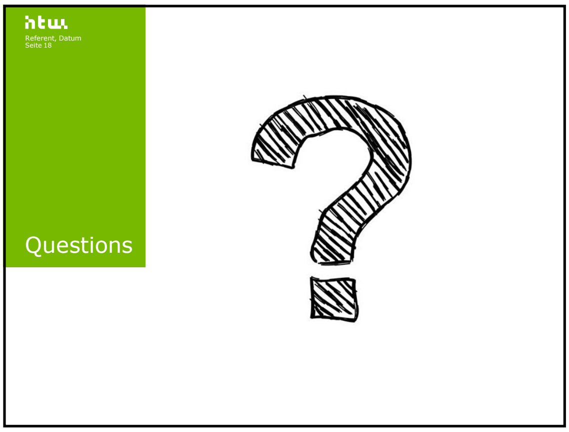
Referent, Datum
Seite 16

PLC
Fieldtest
Modem



Referent, Datum
Seite 17

PLC
Fieldtest
Modem



Referent, Datum
Seite 18

Questions