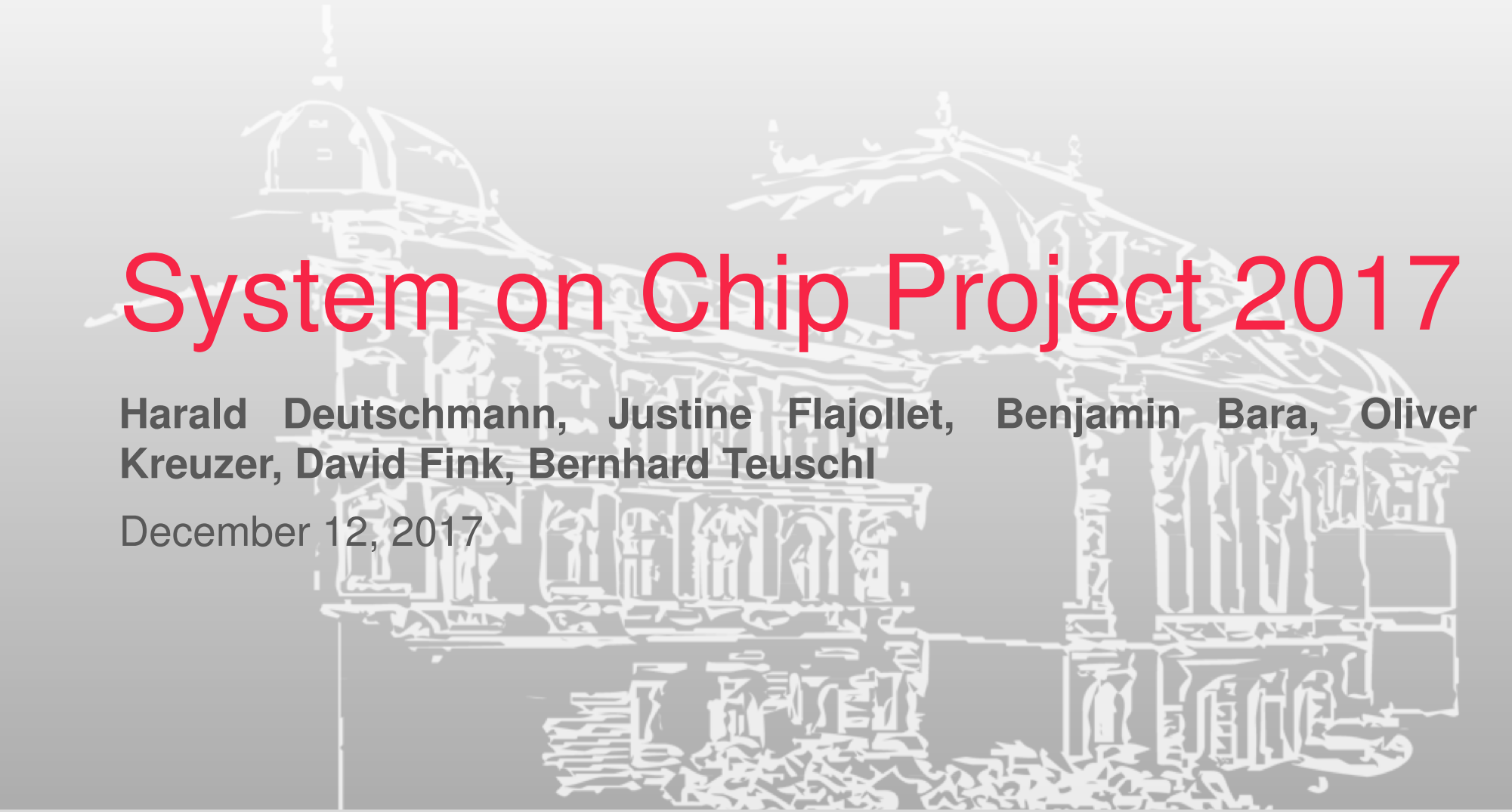


System on Chip Project 2017

Harald Deutschmann, Justine Flajollet, Benjamin Bara, Oliver Kreuzer, David Fink, Bernhard Teuschl

December 12, 2017

Smart Home

Concept

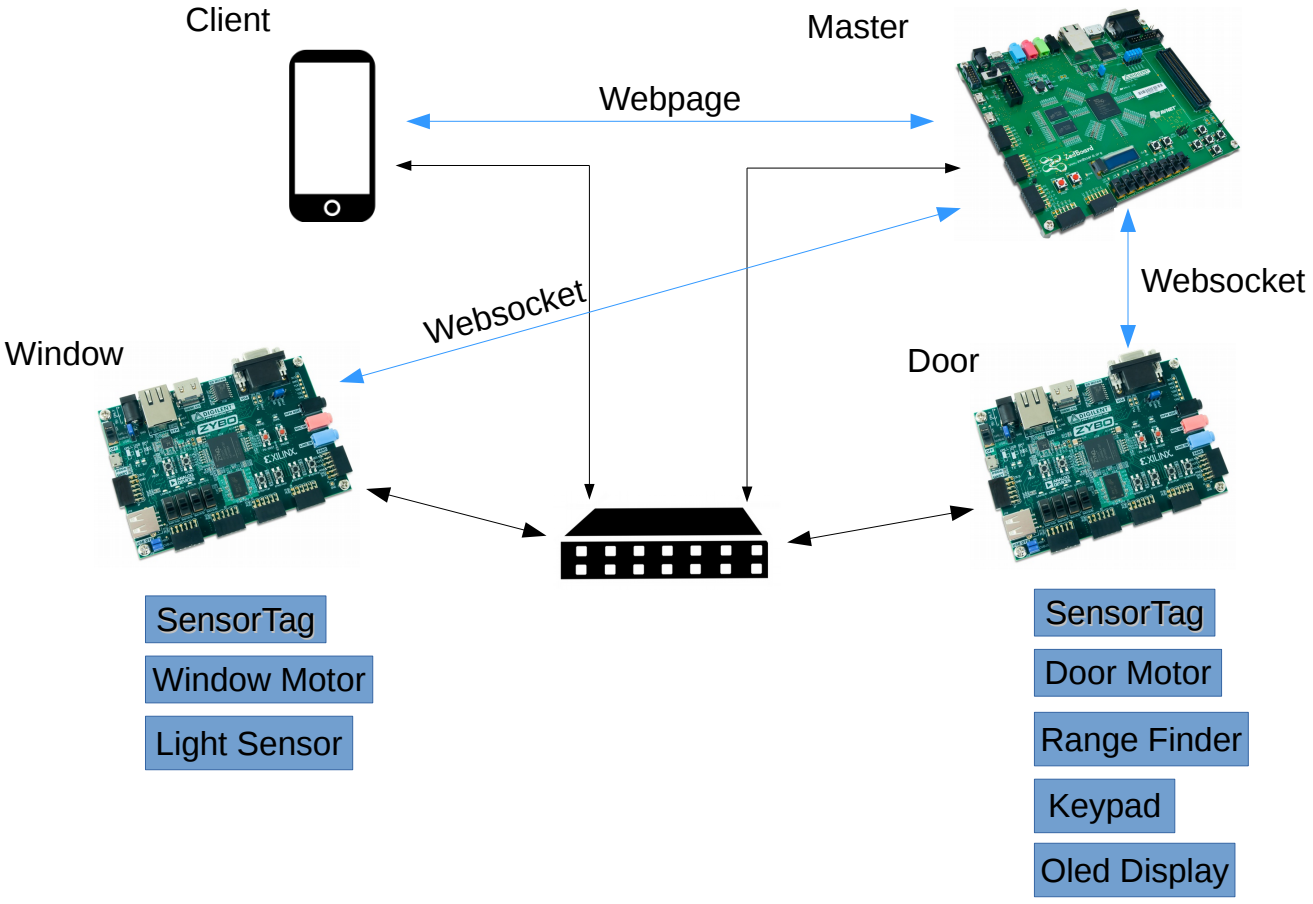
- Sensors to check the weather outside the room
- Sensors to check the climate and lighting inside the room
- Actors to handle the environment
- Access Control
- Web interface

Just like Legos

Building Blocks of the Smart Home System

- One master board to rule them all
- Ethernet
- Slave boards
- Modules - sensors || actors (Pmods)

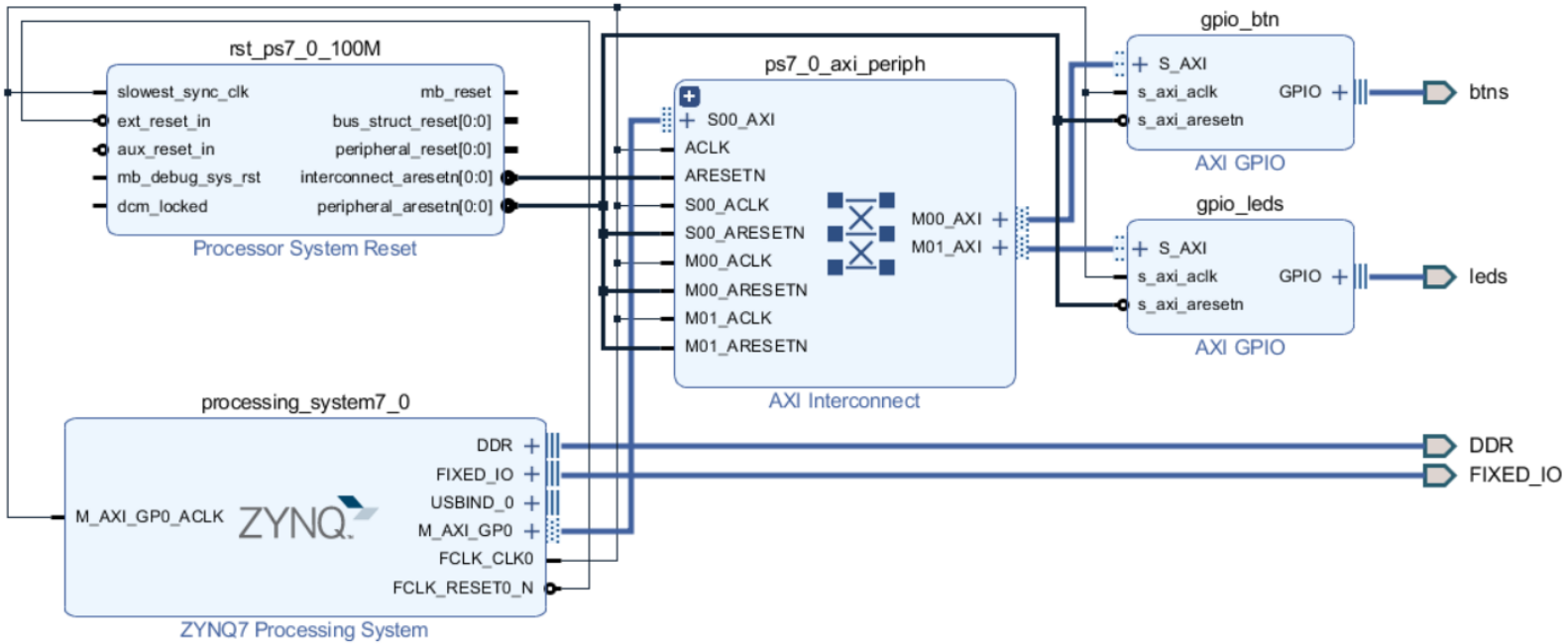
Overview


Client
Master
Webpage
Websocket
Websocket
Window
Door
SensorTag
Window Motor
Light Sensor
SensorTag
Door Motor
Range Finder
Keypad
Oled Display

Master

Node JS application (Master)
IP Address:
192.168.10.1

Master

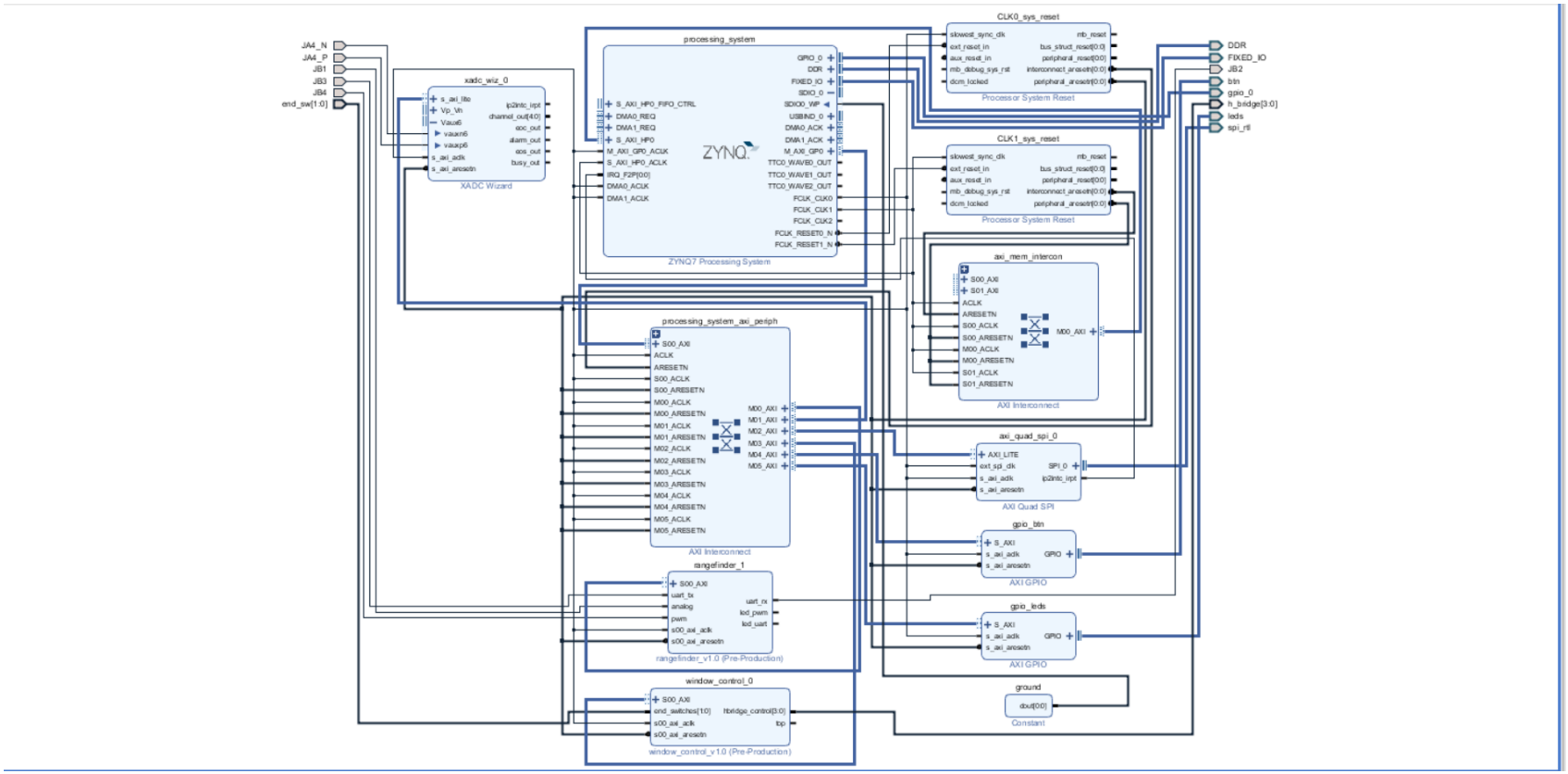

rst_ps7_0_100M
slowest_sync_clk
ext_reset_in
aux_reset_in
mb_debug_sys_rst
dcm_locked
mb_reset
bus_struct_reset[0:0]
peripheral_reset[0:0]
interconnect_aresetn[0:0]
peripheral_aresetn[0:0]
Processor System Reset
ps7_0_axi_periph
S00_AXI
ACLK
ARESETN
S00_ACLK
S00_ARESETN
M00_ACLK
M00_ARESETN
M01_ACLK
M01_ARESETN
M00_AXI
M01_AXI
AXI Interconnect
gpio_btn
S_AXI
s_axi_aclk
s_axi_aresetn
GPIO
AXI GPIO
btns
gpio_leds
S_AXI
s_axi_aclk
s_axi_aresetn
GPIO
AXI GPIO
leds
processing_system7_0
M_AXI_GP0_ACLK
ZYNQ
DDR
FIXED_IO
USBIND_0
M_AXI_GP0
FCLK_CLK0
FCLK_RESET0_N
ZYNQ7 Processing System
DDR
FIXED_IO

Slave: Window

Node JS application (Slave)
IP Address:
192.168.10.2

Modules:	Lightsensor	(working)
	Rangefinder	(working)
	Sensortag	(working)
	Window Motor	(working)

Slave: Window

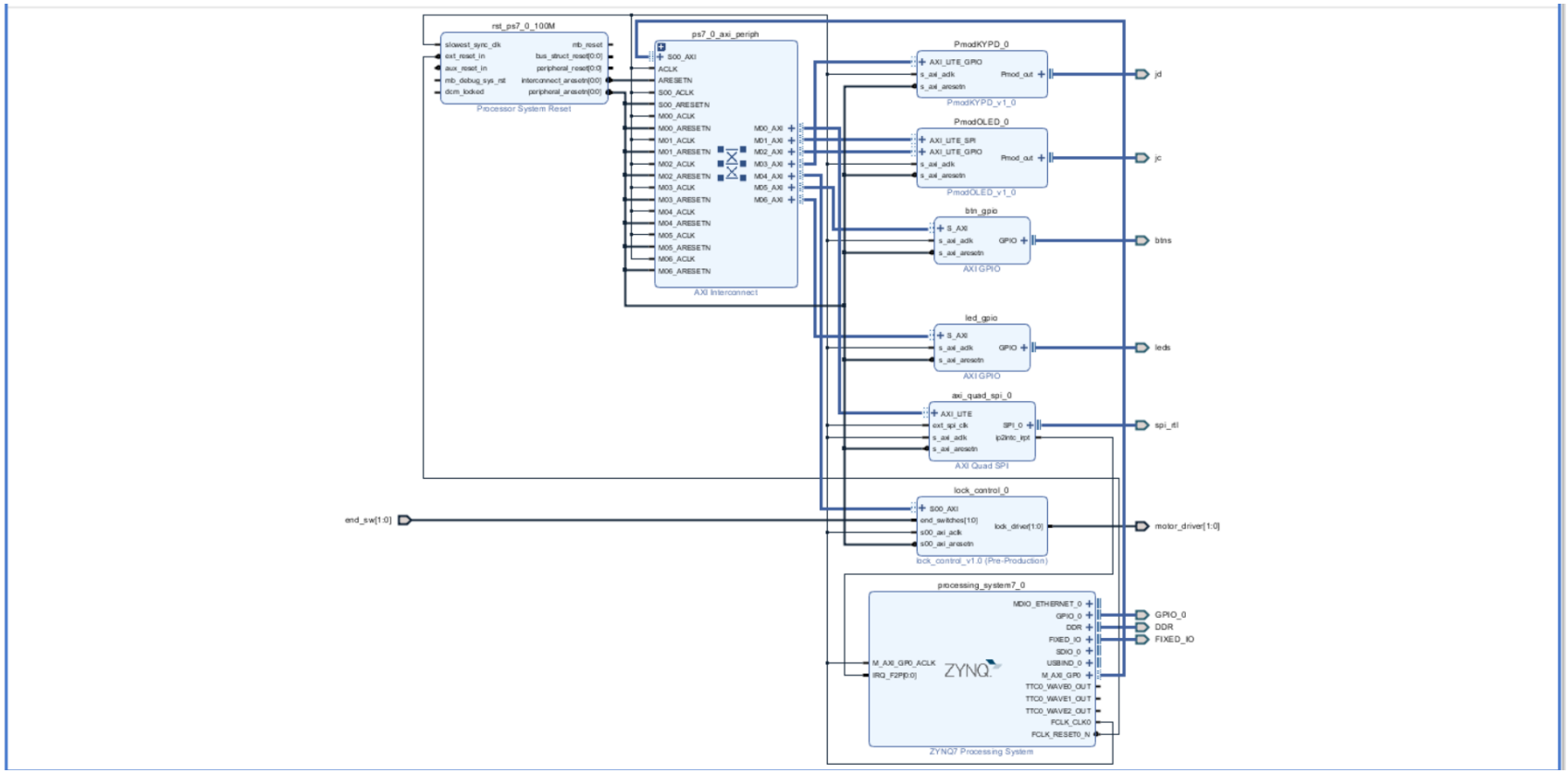


Slave: Door

Node JS application (Slave)
IP Address:
192.168.10.3

Modules:	Rangefinder	(working)
	Sensortag	(working)
	Door Motor	(working)
	OLED-Display	(was planned)
	Keypad	(working)

Slave: Door


rst_ps7_0_100M
Processor System Reset
ps7_0_axi_periph
S00_AXI
AXI Interconnect
PmodKYPD_0
PmodKYPD_v1_0
PmodOLED_0
PmodOLED_v1_0
btn_gpio
AXI GPIO
led_gpio
AXI GPIO
axi_quad_spi_0
AXI Quad SPI
lock_control_0
lock_control_v1.0 (Pre-Production)
processing_system7_0
ZYNQ7 Processing System
end_sw[1:0]
motor_driver[1:0]
jd
jc
btns
leds
spi_rtl
GPIO_0
DDR
FIXED_IO

Node JS: Registering a Slave

- Slave polls static IP address of master
- User has to accept new slave
- Public key exchange between slave and master

Node JS: Registering a Client

- Open the web interface
- Choose a password
- Click Login
- Push the register button

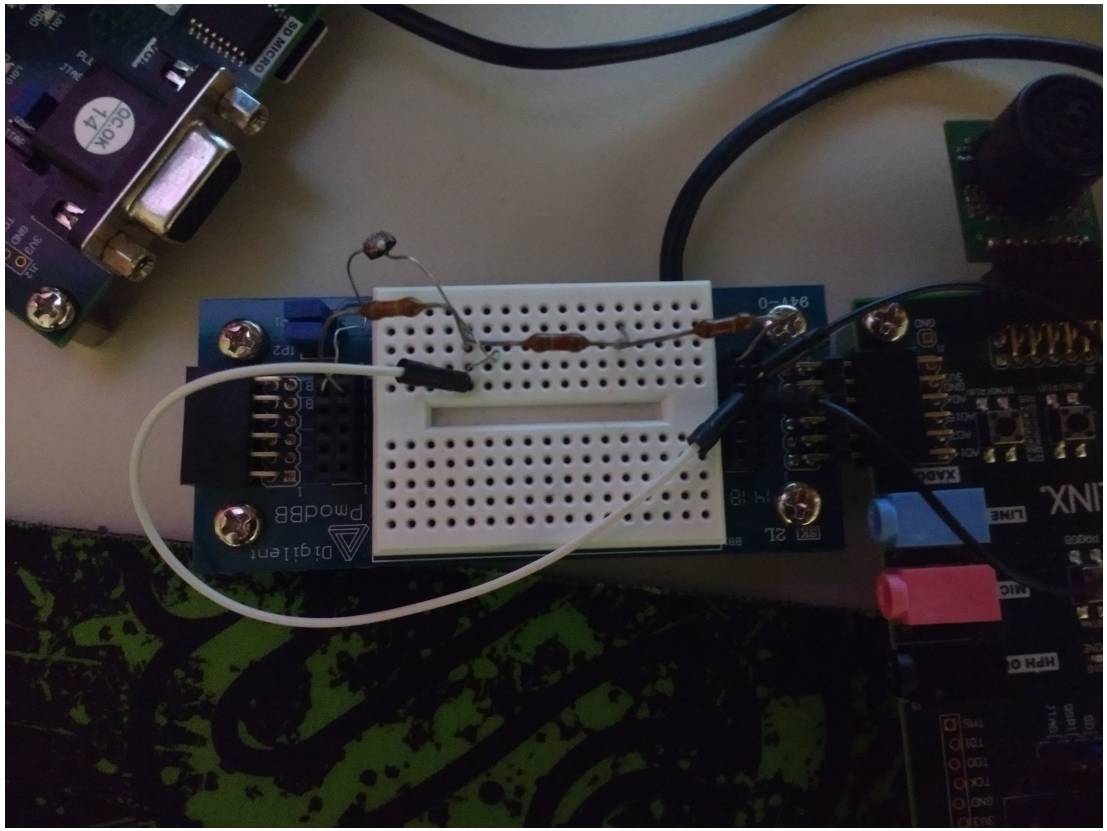
Node JS: Connection Protocol/Security

- WebSockets Secure are used (TLS on TCP) for internal communication
- HTTPS server uses self-signed certificate
- Additional handshake to accept slaves and clients (automatically)
- Public keys are exchanged and verified during handshake
- Still TODO: encryption of file system, bitstream, secure boot.

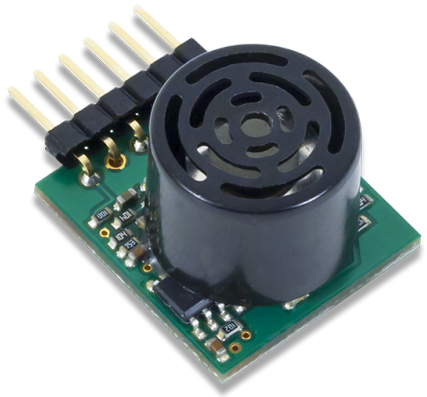
Lightsensor

- Idea: get the voltage (which is relative to luminous intensity) via an ADC
- Values between 50 Ohm (light) and 500 kOhm (dark)
- Supply voltage is 3.3 V
- ADC module allows input between 0 and 1 V
- Voltage divider to get inside this range

Lightsensor

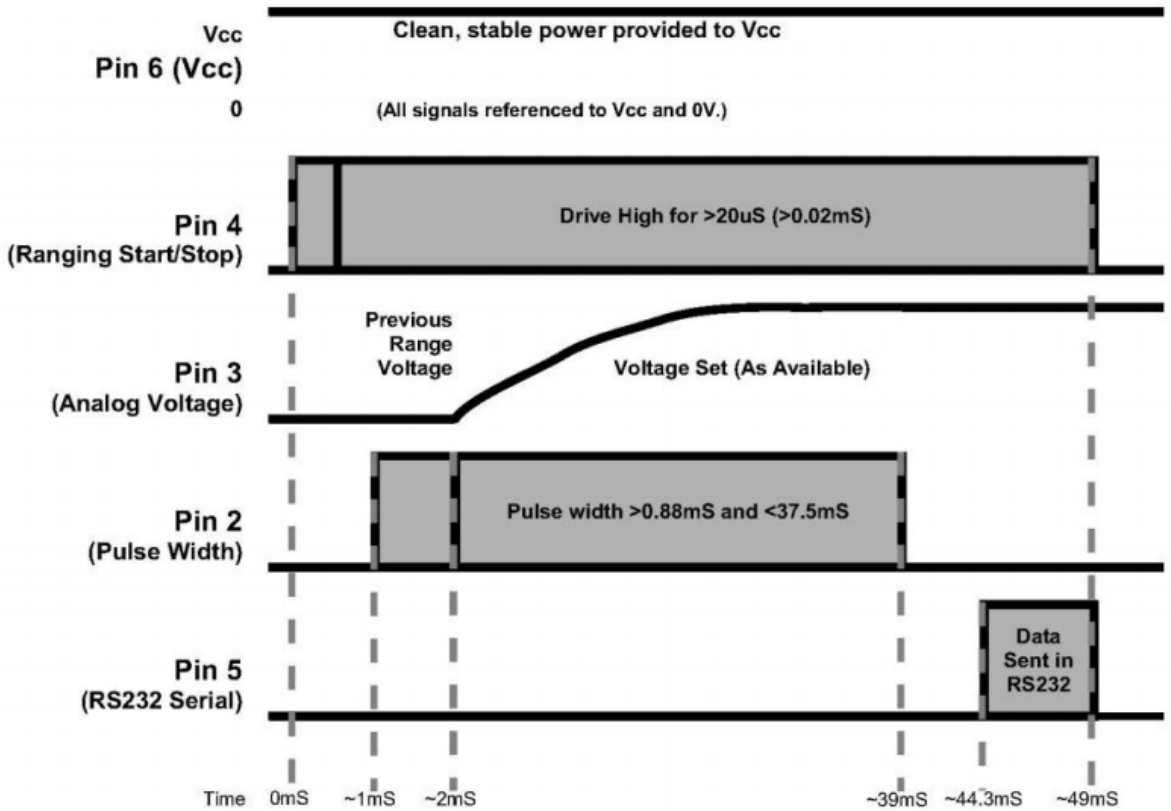


Rangefinder

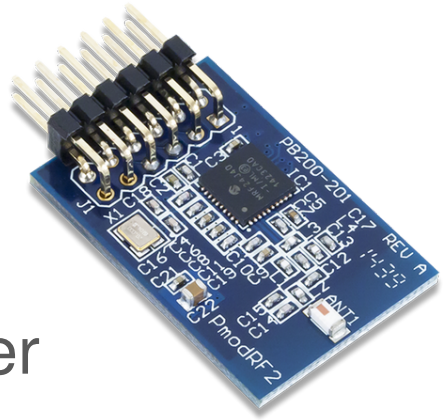


- Three different signals:
  - Analog voltage
  - **PWM**
  - UART
- Two counters as IP core:
  - pulse width
  - whole period
- Maximal distance: 6.50m

Rangefinder

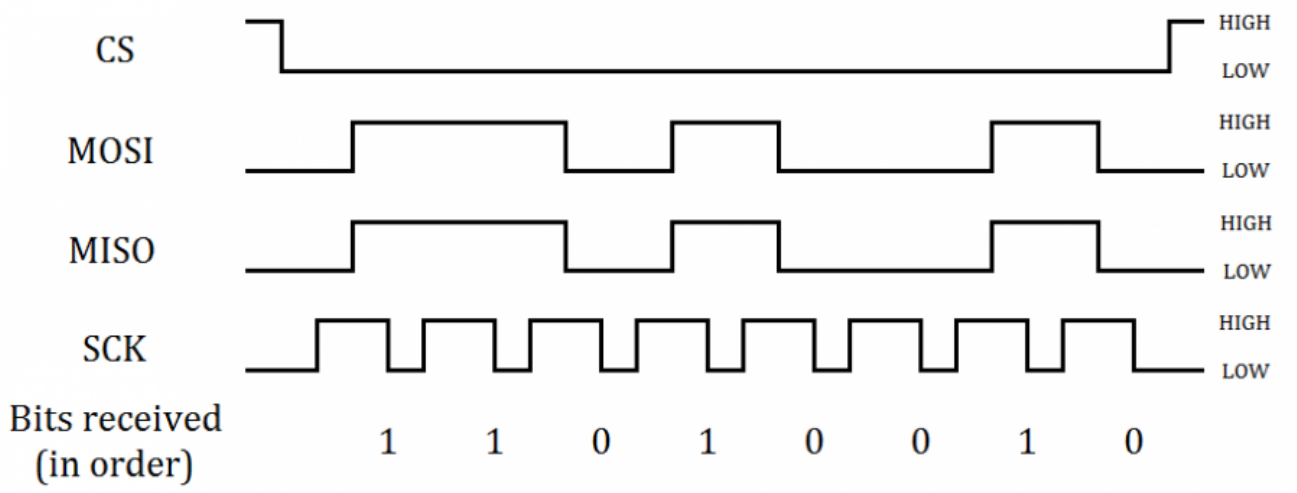


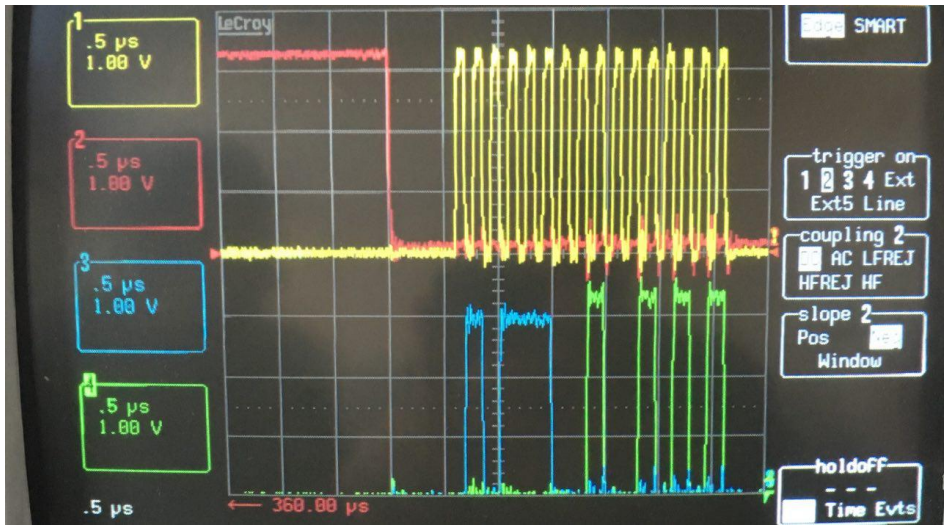
PMOD RF2



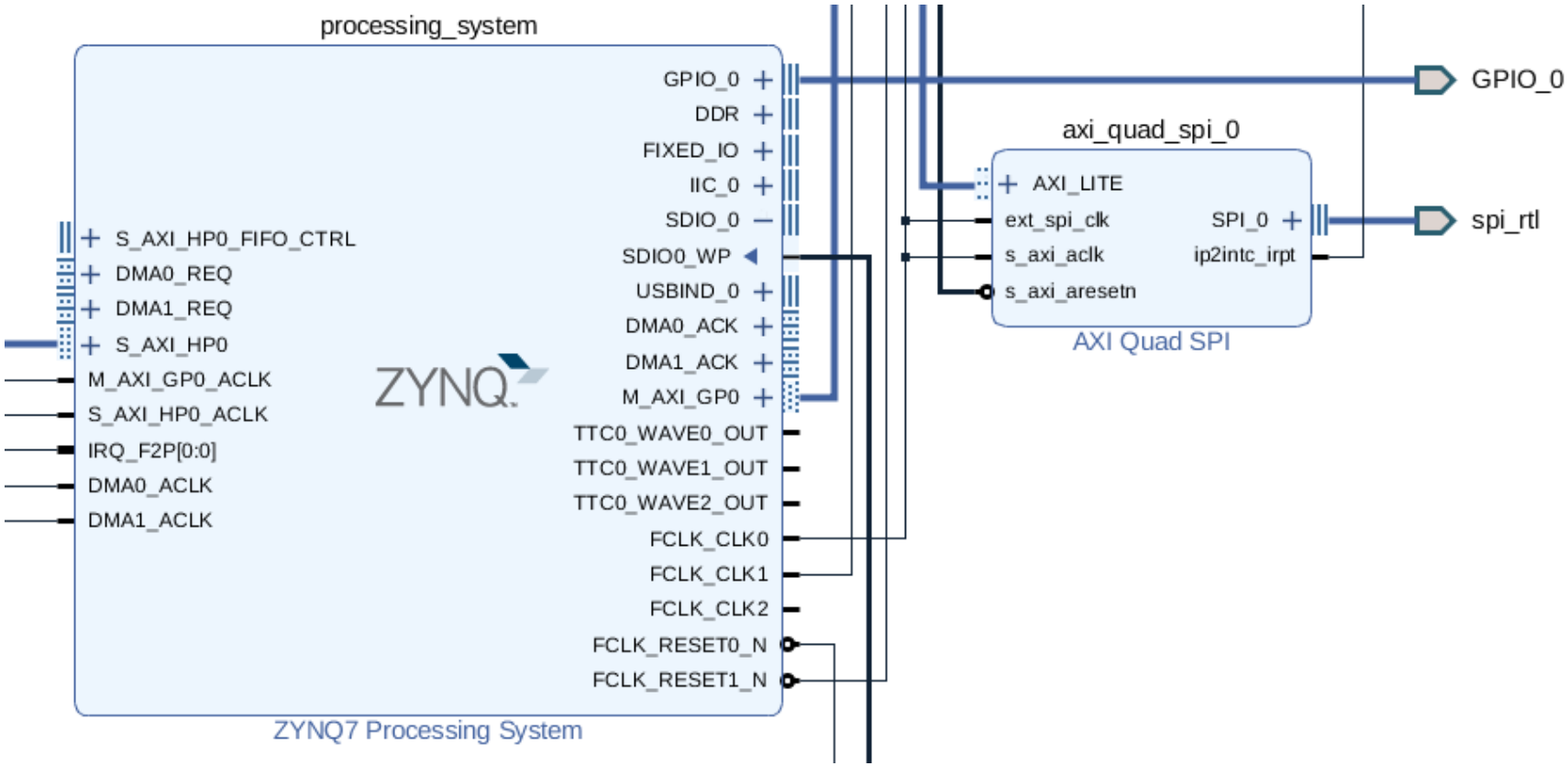
- Microchip MRF24J40 transceiver
- 12-pin Pmod connector
- IEEE 802.15-compliant (supports ZigBee and other wireless networking protocols)
- ISM band 2.405-2.48 GHz operation
- transmits data at speeds up to 625 kbps

SPI Interface





Hardware

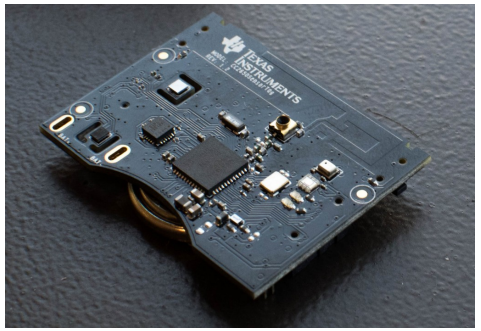

processing_system
S_AXI_HP0_FIFO_CTRL
DMA0_REQ
DMA1_REQ
S_AXI_HP0
M_AXI_GP0_ACLK
S_AXI_HP0_ACLK
IRQ_F2P[0:0]
DMA0_ACLK
DMA1_ACLK
ZYNQ
GPIO_0
DDR
FIXED_IO
IIC_0
SDIO_0
SDIO0_WP
USBIND_0
DMA0_ACK
DMA1_ACK
M_AXI_GP0
TTC0_WAVE0_OUT
TTC0_WAVE1_OUT
TTC0_WAVE2_OUT
FCLK_CLK0
FCLK_CLK1
FCLK_CLK2
FCLK_RESET0_N
FCLK_RESET1_N
ZYNQ7 Processing System
axi_quad_spi_0
AXI_LITE
ext_spi_clk
s_axi_aclk
s_axi_aresetn
SPI_0
ip2intc_irpt
AXI Quad SPI
GPIO_0
spi_rtl

Driver

- Linux mainline driver for MRF24J40 available
- Board needs devicetree support
- GPIO EMIO for interrupts from the microchip

```
axi_quad_spi_0: axi_quad_spi@41e00000 {
  bits-per-word = <8>;
  compatible = "xlnx,xps-spi-2.00.a";
  fifo-size = <16>;
  interrupt-parent = <&intc>;
  interrupts = <0 29 1>;
  num-cs = <0x1>;
  reg = <0x41e00000 0x10000>;
  xlnx,num-ss-bits = <0x1>;
  xlnx,spi-mode = <0>;
  #address-cells = <1>;
  #size-cells = <0>;
  mrf24j40@0 {
    status = "okay";
    compatible = "mrf24j40";
    spi-max-frequency = <5000000>;
    reg = <0>;
    interrupts = <0 20 4>;
    interrupt-parent = <&gpio0>;
  };
};
```

TI Sensortag



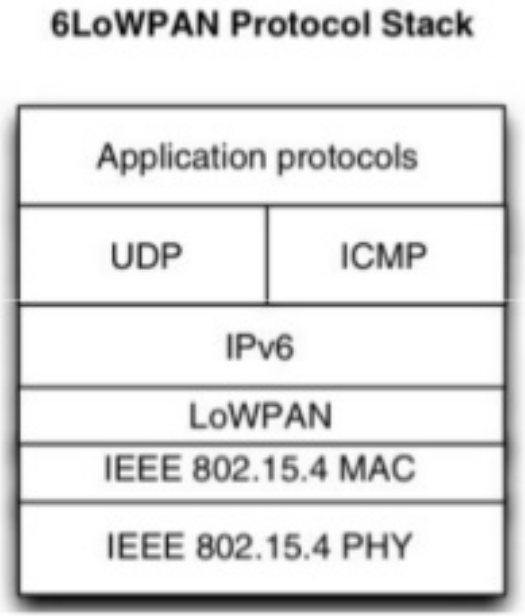
- Cortex-M0 RF core and Cortex-M3 user application core
- 10 sensors (temperature, pressure, humidity, light, gyro, accelerometer, microphone, ...)
- Two buttons, two LEDs, buzzer
- Supports Bluetooth low energy, ZigBee, 6LoWPAN
- Runs an application in Contiki: The Open Source OS for the Internet of Things

IEEE 802.15.4

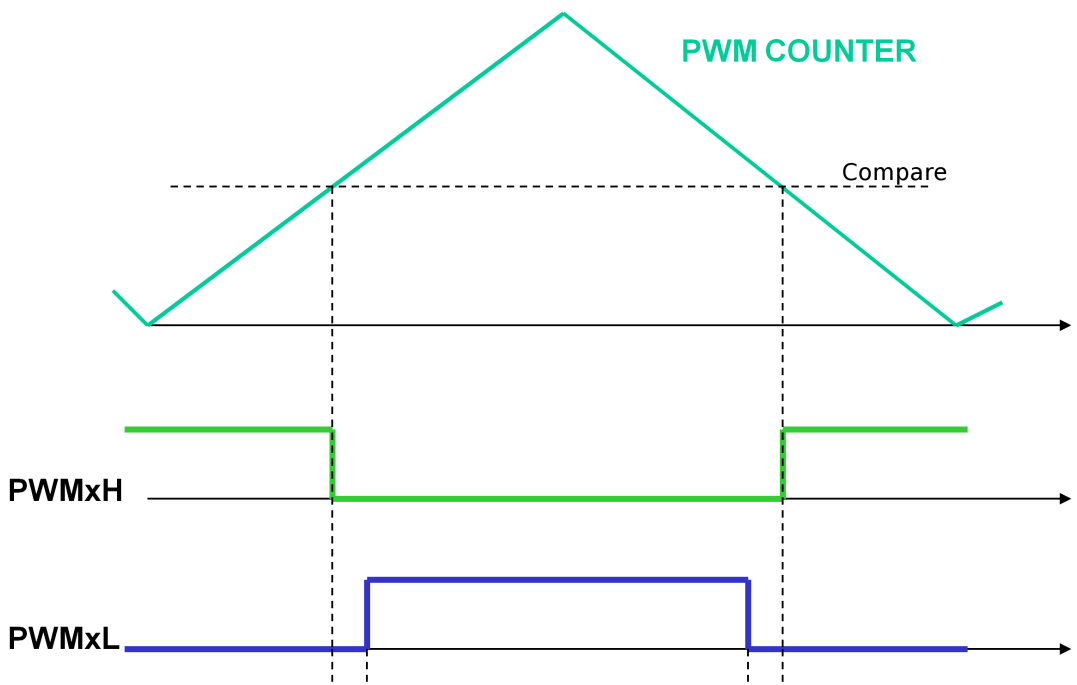
- Specifications for low-rate, low-power wireless networks
- Designed for small sensors to run months/years on battery
- 127 bytes MTU and 250 kbit/s
- Sometimes confused with ZigBee as it is used as PHY and MAC layer there

6LoWPAN

- IPv6 over LoWPAN (IEEE 802.15.4)
- Adaption layer between data-link and network layer
- Frame encapsulation and fragmentation
- Efficient header compressions



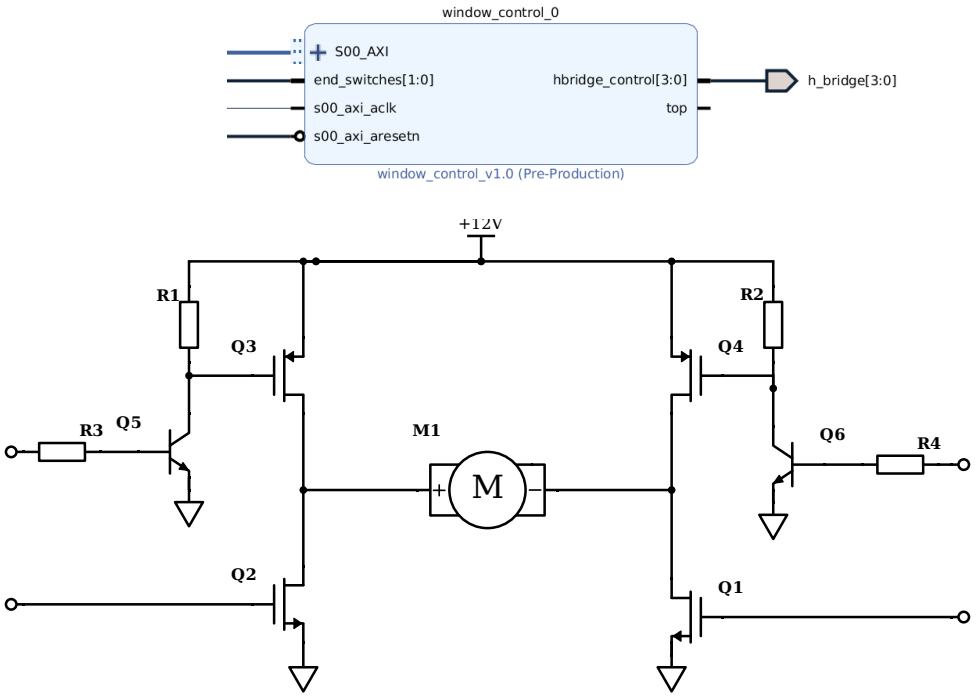
Window: Motor Centroid PWM



$$TopValue = 5000 \qquad f_{pwm} = \frac{f_{in}}{Top \cdot 2} = 10kHz$$

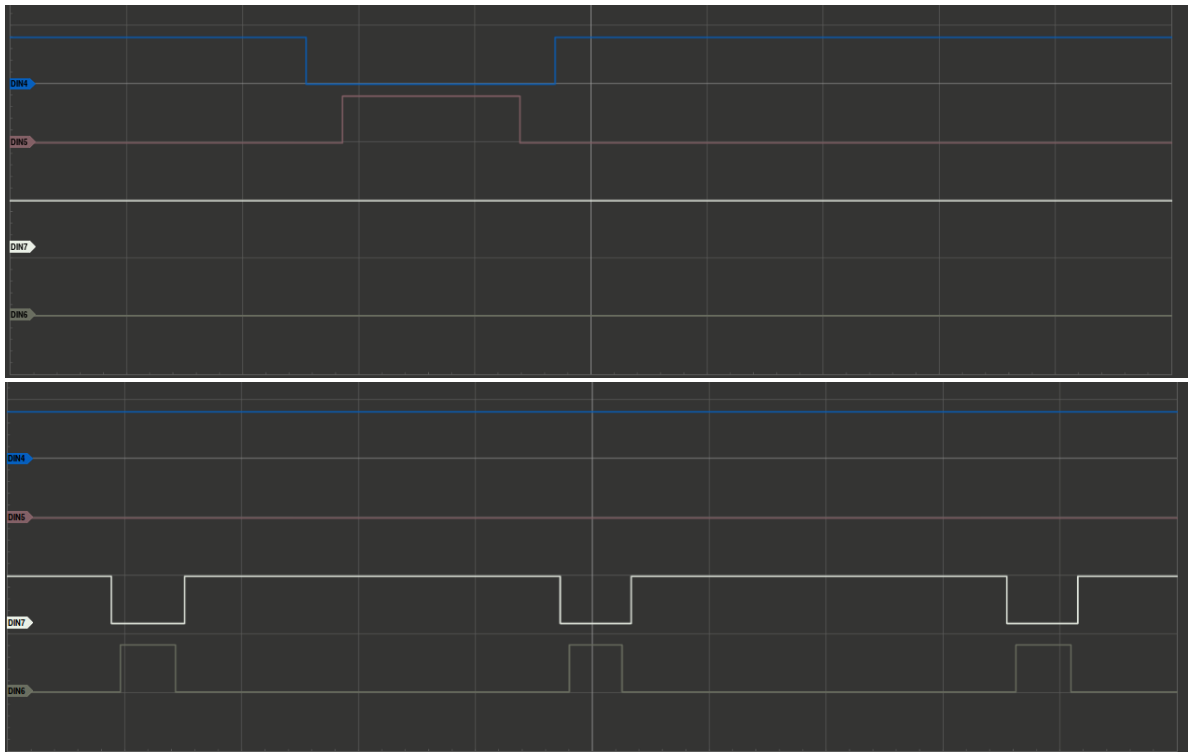
Window: Motor H-Bridge Schematic

Self built driver for motor control.



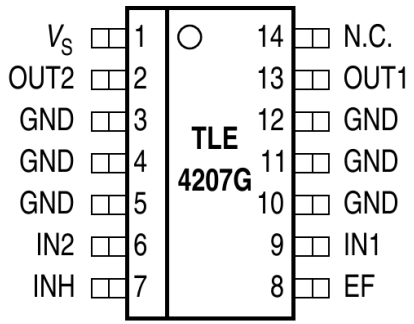
Window: Motor H-Bridge Driver Signals

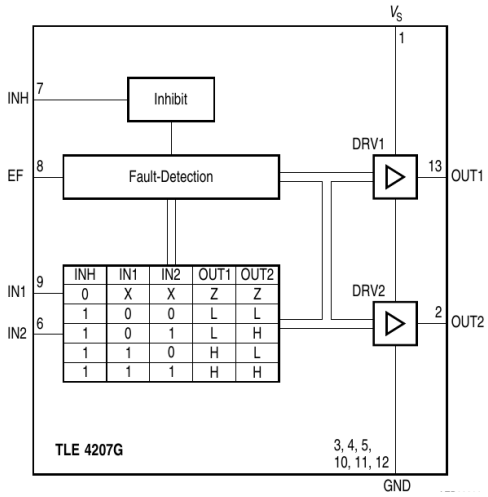
Input signals of the H-Brigde driver for different motor directions.



Door: Door Motor

Used for motor control Infineon TLE4207G



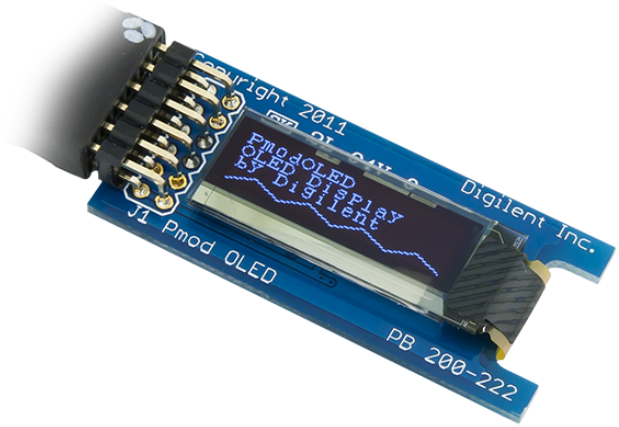


INH	IN1	IN2	OUT1	OUT2
0	X	X	Z	Z
1	0	0	L	L
1	0	1	L	H
1	1	0	H	L
1	1	1	H	H

Functional Truth Table

INH	IN1	IN2	OUT1	OUT2	Mode
0	X	X	Z	Z	Stand-By
1	0	0	L	L	Brake LL
1	0	1	L	H	CW
1	1	0	H	L	CCW
1	1	1	H	H	Brake HH

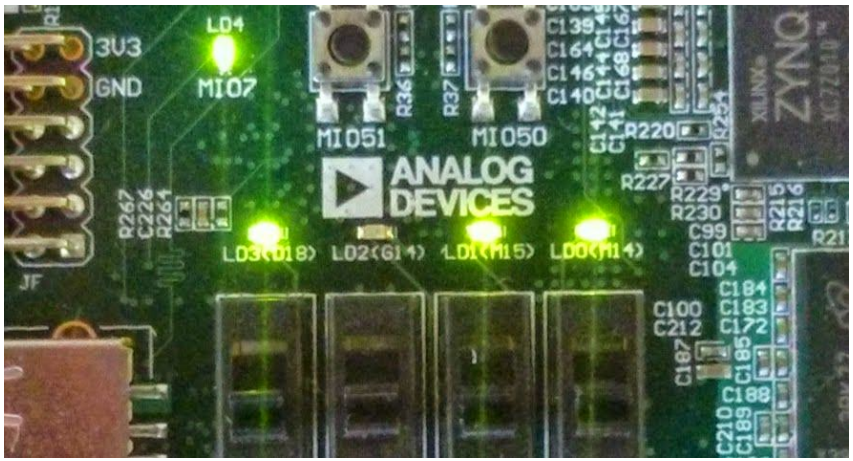
Door: OLED-Display



- Was working on bare-metal
- Conflict with SPI
- Driver problems....

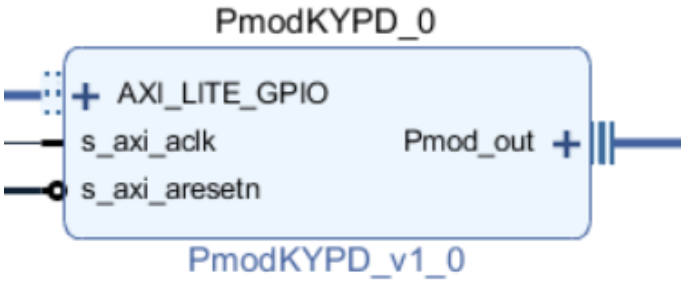
Door: LEDs

Solution:
Display through LEDs!



Door: Keypad

Xilinx IP-Core - Custom Driver



Door Access - Application

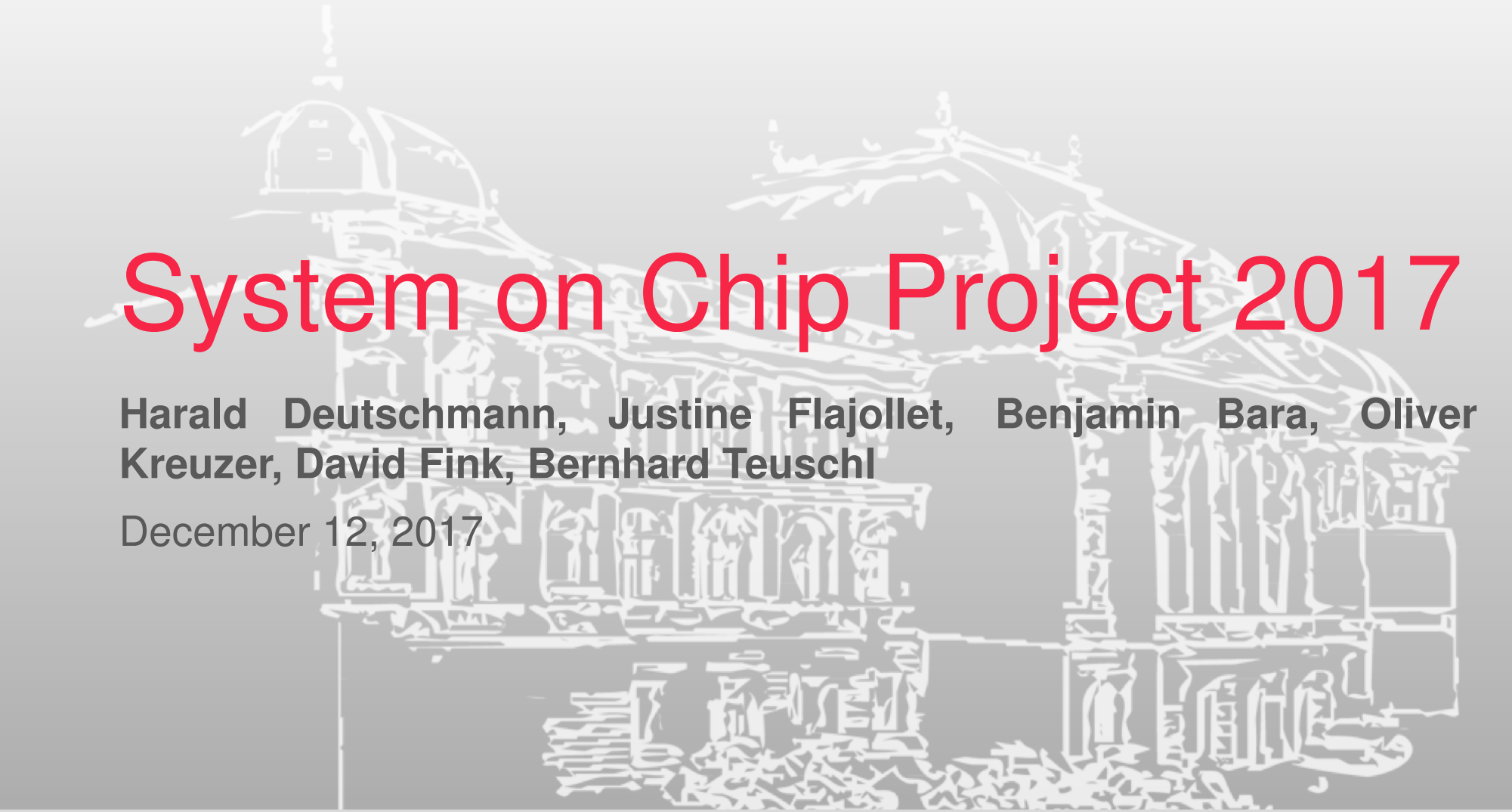
- System password set on first launch
- 3 options: Key, Enter and Delete
- Limited number of tries (LEDs)
- Too long idle: timeout

Keypad - Driver

- proc file
- outputs the pushed key
- filters for matrix errors
- filters for constantly pushed key

Demo

Live Demo



System on Chip Project 2017

Harald Deutschmann, Justine Flajollet, Benjamin Bara, Oliver Kreuzer, David Fink, Bernhard Teuschl

December 12, 2017