h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

Software Development Assignment
Lab 3

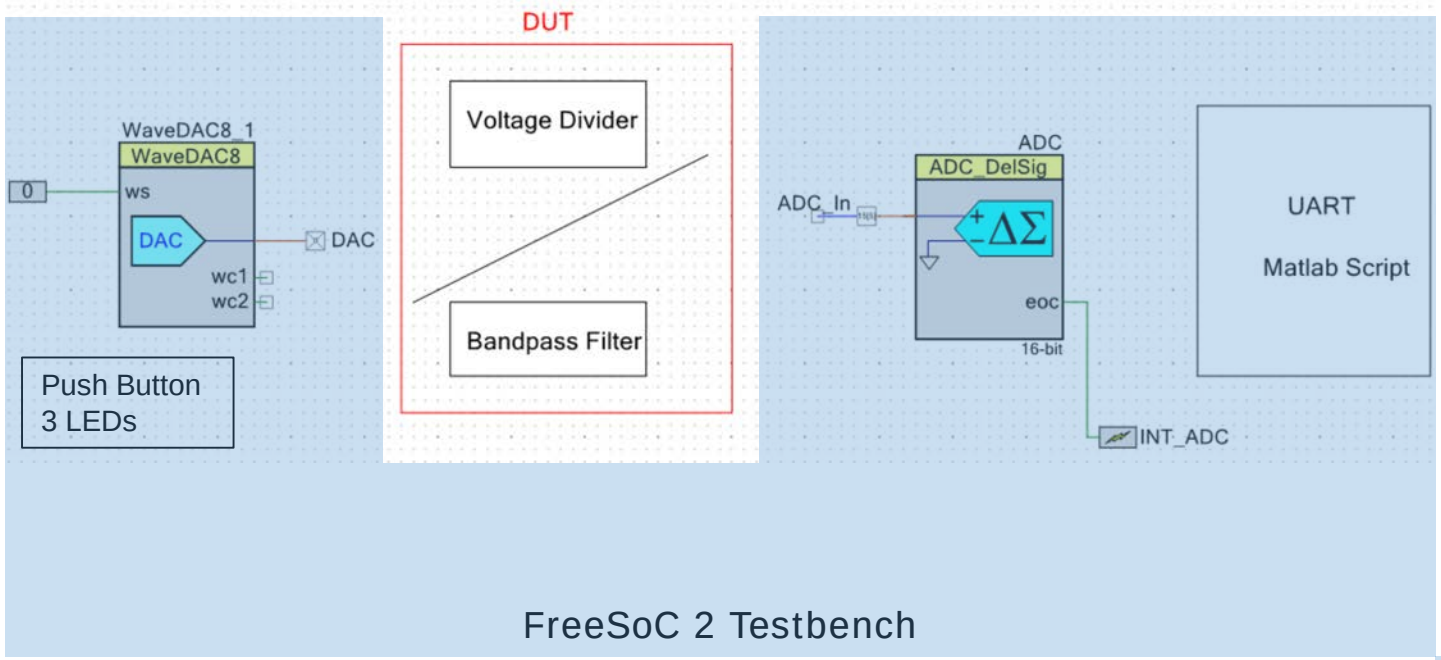
Systen Driven Hardware Design

19.05.22

Prof. Dr.-Ing. Stephan Bannwarth

stephan.bannwarth@h-da.de

Testbench with FreeSoc 2



FreeSoC 2 Testbench

Idea Behind

Use the FreeSoC 2 as a testbench for your Amplifier on the Arduino shield.

Do three steps

1st Step

In order to test your software, just use a simple voltage divider between DAC and ADC.

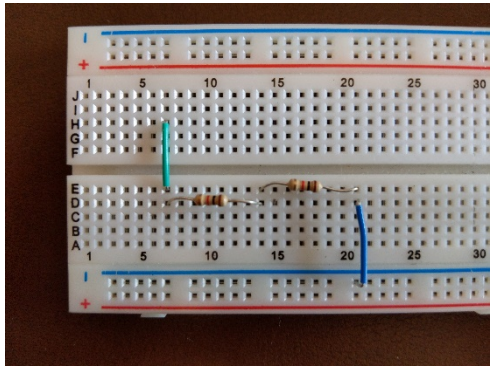
2nd Step

Exchange the voltage divider by your bread board amplifier

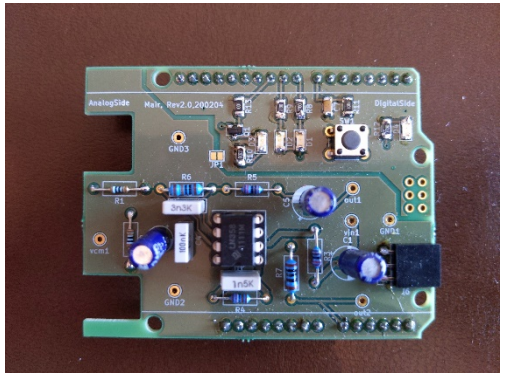
3rd Step

Excahange the breadboard by your Arduino shield

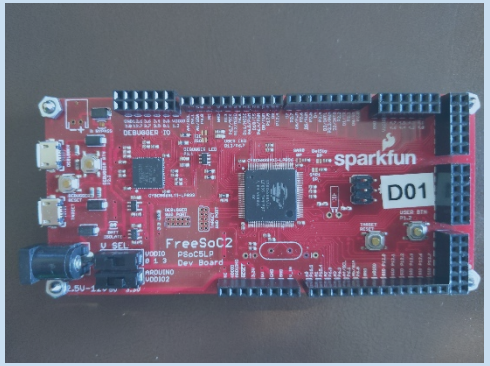
→ last Lab.



Testing the Testbench



Testing your PCB



FreeSoc2 Testbench

```
% Behavior:
% - Everytime Maltlab writes ‘s’on the UART, the PSoC sends new
measurement
%    results and Matalab writes 'o' if these data is received.
% - The Script terminates after 10 data transfers.
%
% Using:
% 1. Connect FreeSoc2 to USB (i.e. Power Up)
% 2. Check the correct serial Port Settings
% 3. Start this Matlab Script
% 4. Run this Script
% 5. Press the external Push Button to start measuring
```

Data Transfer FreeSoc 2 and Matlab: *Testebench_COM2Matlab.m* excerpt

```
flg_data_avai = 0;
fwrite(PSoC,'s','uchar') % means send, I am ready to receive
while(flg_data_avai == 0)

        if PSoC.BytesAvailable == 2048
              fwrite(PSoC,'o','uchar') % means I received all expected data
              rx_data_adc = fread(PSoC,1024,'uint16');
              fprintf(" Transfer %i DONE \n",count);

              % Plotting the received data
              figure(f1)
              subplot(2,1,1)
              plot([0:(length(rx_data_adc)-1)],rx_data_adc(1:(length(rx_data_adc))));
              title(['Received Time Domain Data No.:',num2str(count)]);
              subplot(2,1,2)
              plot([0:1023],1/length(rx_data_adc)*20*log10(abs(fft(rx_data_adc))));
              title('FFT -  Matlab');

              % Save the received data
              save(strcat('CW_rx_data_adc_',int2str(count),'.mat'),'rx_data_adc');
              count=count+1;
        end

        if count == 11
            break;
        end

        fwrite(PSoC,'s','uchar') % means send, I am ready to receive
end
```

N.B. The complete script can be found in moodle

Software Development Assignment

Aim

Use a DAC (e.g. waveform DAC), which provides the test signal. Use a 16-bit SD-ADC to sample the voltage divider output and store the samples in an array. If Matlab requests by UART, send the ADC sampled data by UART to Matlab. This is repeated 10 times before the Matlab Script terminates, please check the Matlab code.

Constraints

- DAC out Pin: **P15[4]**
- ADC in Pin: **P15[5]**
- Push-Button: **P2[4]**
- LEDs: green **P2[5],** orange **P2[6],** red **P2[7]**
- Your code must work with the “Testebench_COM2Matlab.m” since we test your compiled code with this script.
- Use a switch-caste structure for state machine (sequencing), see slide “Sequencing”

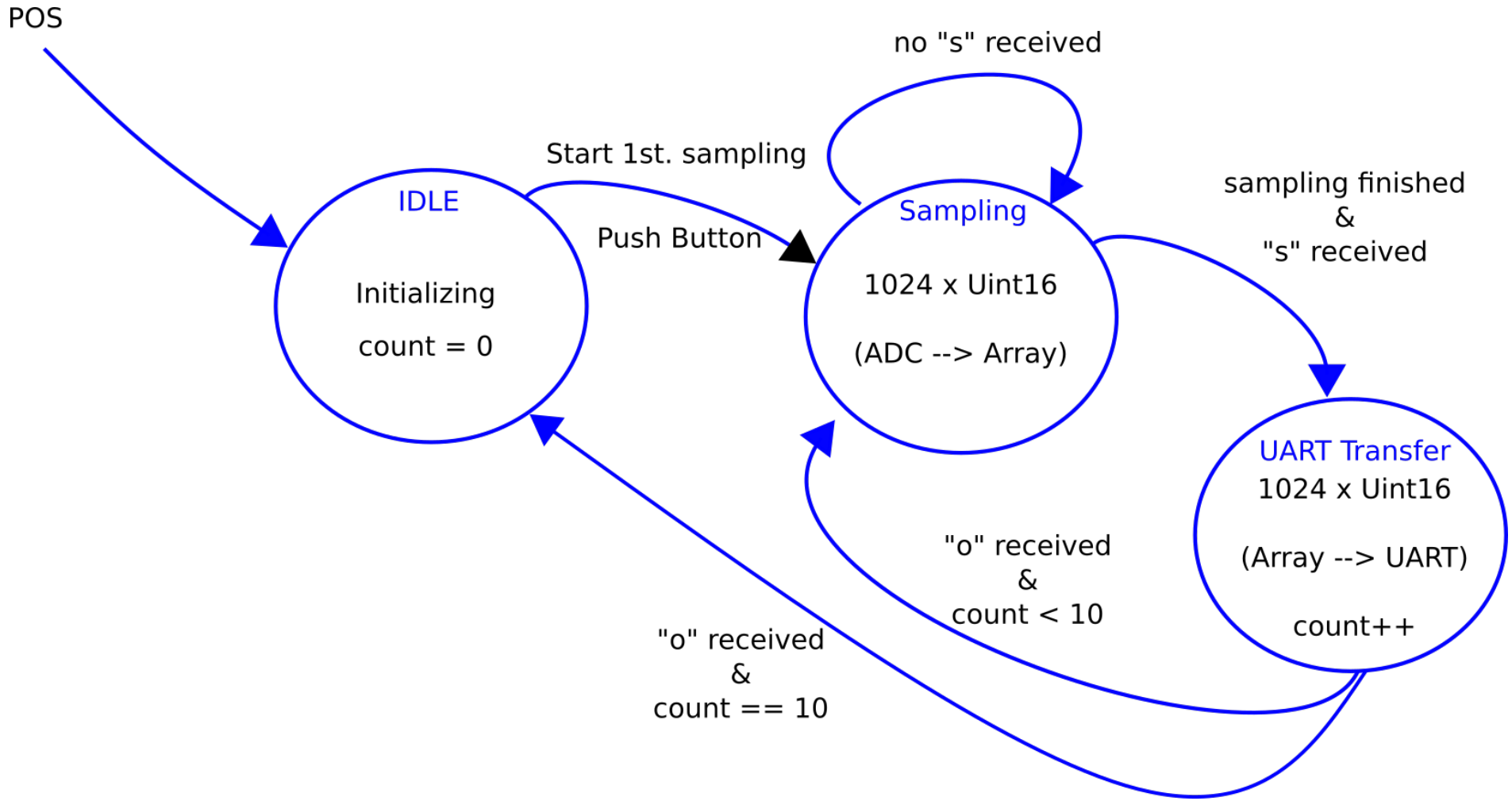
Deliverables

- PSoC Project
- Compiled and linked program, ready to run
- pdf including:
  1. State Machine Diagram for the PSoC C-Programm
  2. Photo of your testbench (bread board, voltage divider, connection to the FreeSoc 2)
  3. Ideas for further improvements

Hints

Before you start with the assignments do the Exercises from the lecture (Heart Beat, 3-Bit Counter, ADC-Reading)

Sequencing



Use a switch-case structure to implement the state machine

Software Development – Scoring Scheme

No Copy and Paste From Others: 0%.

40%: Code can be flashed and is running,
For example and not limited to

Code compiles and can be flashed
Given DAC out and ADC in Ports
Given Voltage Divider
After flashing the data arrives in our Matlabscript

20%: Software Architecture and PSoC Configuration, for example and not limited to

Polling / Interrupts / use of global flags
functions have been used to make the code modular

20%: Clean Software Design with minor issues, for example and not limited to

Clear variable names
Swtich-case
Interrupts fairly used
Test benches for used functions
Useful comments
Readability

10%: Your ideas beyond the said and further considerations

10%: Presentation and Questions