MPT3

Mikroprozessortechnik

HSD 3. Semester

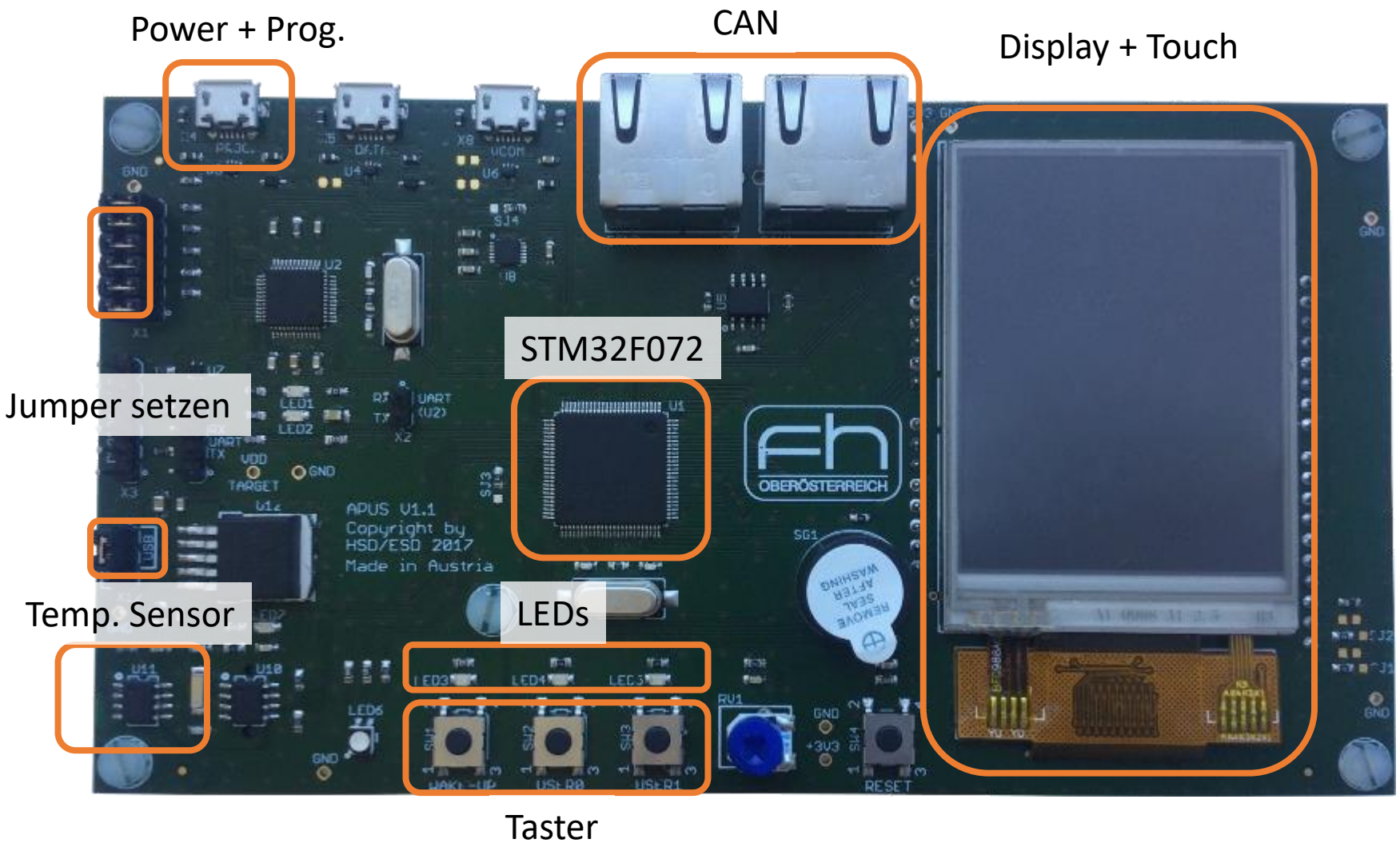
Aufbau und Ziele

- 9 Übungen
- Eval Board APUS
  - STM32F072VBT6 Prozessor
  - 32 Bit ARM Cortex-M0, 48 MHz Takt
  - 128 kB ROM, 16 kB RAM, 87 GPIO-Pins
- Programmierung in C für Mikrocontroller
- Datenblätter lesen
- Schaltplan Entwicklung

Inhalte

- Kennenlernen Eval-Board
- Erste Schritte bei der Programmierung
- I/O Ansteuerung (GPIO)
  - LEDs, Taster
  - Lauflicht, Alarmanlage
- Interrupts & NVIC
- Clock-Tree & SysTick
- Timer, UART, ADC

Eval-Board APUS



Eval-Board APUS

- STM32F072VB
- Keil uVision5 (IDE)
- ST-LINK/V2 (Programmer, Debugger)
- Features
  - LEDs, Buttons, Potentiometer
  - UART, VCOM, USB, CAN
  - Newhaven TFT Display with integrated Touch Panel
  - Temperature Sensor, Real Time Clock
  - Arduino Shield Connector, Piezo Buzzer

Datenblätter

- STM32F0x2 Reference manual.pdf
  - Ansteuerung Peripherie, Adressen, Register
- STM32F072x8 Datasheet.pdf
  - Alternative Portfunktionen, Electrical characteristics
- Cortex M0 Technical Reference Manual.pdf
  - Allgemeine Cortex-M0 Infos & Register

Start IDE

- Keil uVision 5
- Device Family Pack „Keil:STM32F0xx_DFP:1.x.0“ installieren (über Pack Installer)
- Schaltplan und Datenblatt einlesen
- Demo-Projekt „Blinky“ öffnen (zuerst kopieren)
  - Single-Step durch das Projekt
- **Wichtig**: vor Zugriff auf Register muss Clock (RCC) aktiviert werden!

Bit-Operationen

- Einzelnes Bit im Register setzen (z.B. Bit 6)
  - tempReg |= (1<<6);
    - tempReg: `xxxx xxxx`
    - (1<<6): `| 0100 0000`
    - Ergebnis: `x1xx xxxx`
- Mehrere Bits im Register setzen (z.B. Bit 2 und 3)
  - tempReg |= (3<<2); ODER
  - tempReg |= ((1<<2) | (1<<3));
    - tempReg: `xxxx xxxx`
    - (3<<2): `| 0000 1100`
    - Ergebnis: `xxxx 11xx`

Bit-Operationen

- Einzelnes Bit im Register löschen (z.B. Bit 4)
  - tempReg &= ~(1<<4);
    - tempReg: `xxxx xxxx` (1<<4: `0001 0000`)
    - ~(1<<4): `& 1110 1111`
    - Ergebnis: `xxx0 xxxx`
- Mehrere Bits im Register löschen (z.B. Bit 4 und 5)
  - tempReg &= ~(3<<4); ODER
  - tempReg &= ~((1<<4) | (1<<5));
    - tempReg: `xxxx xxxx` (3<<4: `0011 0000`)
    - ~(3<<4): `& 1100 1111`
    - Ergebnis: `xx00 xxxx`

Bit-Operationen

- Einzelnes Bit im Register toggeln (z.B. Bit 3)
  - tempReg ^= (1<<3);
    - tempReg: `xxxx 1xxx`
    - (1<<3): `^ 0000 1000`
    - Ergebnis: `xxxx` **`0`**`xxx`

    - tempReg: `xxxx 0xxx`
    - (1<<3): `^ 0000 1000`
    - Ergebnis: `xxxx` **`1`**`xxx`

Bit-Operationen

- Prüfen, ob ein einzelnes Bit im Register gesetzt ist (z.B. Bit 3)
  - `if (reg & (1<<3)) { ... }`

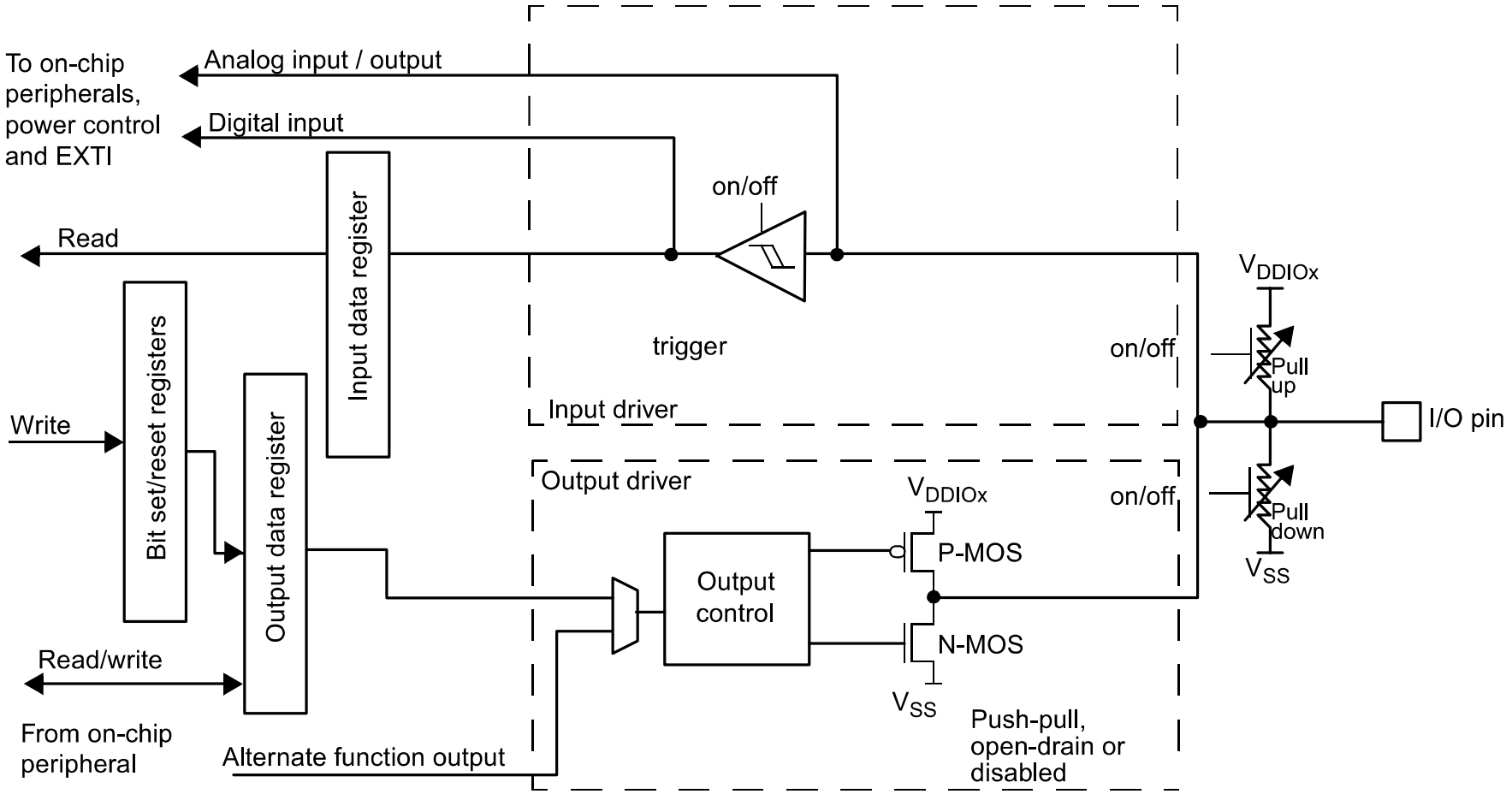
- Prüfen, ob mehrere Bits im Register gesetzt sind (z.B. Bit 3 und Bit 5)
  - `#define BITMASK  ((1<<3) | (1<<5))`
  - `if ((reg & BITMASK) == BITMASK) { ... }`
    - Liefert nur dann TRUE, wenn alle Bits von BITMASK in reg gesetzt sind

Bit-Operationen

- Prüfen, solange ein einzelnes Bit im Register NICHT gesetzt ist (z.B. Bit 3)
  - `while ((reg & (1<<3)) == 0) { ... }`

- Prüfen, solange bestimmte Bits im Register NICHT gesetzt sind (z.B. Bit 6 und Bit 12)
  - `#define BITMASK  ((1<<6) | (1<<12))`
  - `while ((reg & BITMASK) != BITMASK) { ... }`
    - Warten, bis alle Bits von BITMASK in reg gesetzt sind

Aufbau GPIO


To on-chip peripherals, power control and EXTI
Analog input / output
Digital input
on/off
Read
Input data register
trigger
Input driver
Write
Bit set/reset registers
Output data register
Output driver
Output control
P-MOS
N-MOS
Read/write
From on-chip peripheral
Alternate function output
Push-pull, open-drain or disabled
V_{DDIOx}
V_{SS}
on/off
Pull up
Pull down
I/O pin

GPIO Register

- STM32F0x2 Reference manual.pdf / Kapitel 9
- Wichtige Register
  - GPIOx_MODER: input, output, analog, alternate
  - GPIOx_ODR, GPIOx_IDR: input/output
- Sonstige Register
  - GPIOx_PUPDR: pull-up, pull-down
  - GPIOx_BSRR: input/output
  - GPIOx_OTYPER: push-pull, open drain
  - GPIOx_OSPEEDR: speed
  - GPIOx_LCKR: lock configuration
  - GPIOx_AFRL, GPIOx_AFRH: alternative Portfunktion