EmbeddedSystems

Repo for Embedded Systems lecture project. (This software is developt for a specific hardware)

Known issues

- ...

Hardware setting (new):

- CPU: ATMEGA1284P (40 pin)
- Programmer: diamex ISP-PRog-NG
- some leds and buttons

Hardware setting (old):

- CPU: ATMEGA328P U (28 pin)
- Programmer: diamex ALL-AVR-PROG
- some leds and buttons

Software setting:

- ATMEL STUDIO 7
- standard config
- disabled optimization

About the project

In this project we want to develop an embedded system based on an ATMEGA1284P.

Software

GPIO

The gpio functions help handling the gpios. The board has 4 buttons and 8 leds for user purpose.

There are to init functions:

- Led_init(int initD)
- Taster_init()

to setup the PORTs correctly. If you are using the ES-Board, set the prarmeter initD to 0. If you are using the old hardware setting, set this parameter to 1. (traffic light uses leds at PORTD)

After setup the button states can be accessed by functions: TasterX_get(), where X is between 1 and 4.

The leds can be controlled with:

- LedX_On()
- LedX_Off()

where X is between 1 and 8.

Timer

Timer0 triggeres an TIMER0_COMPA_vect interrupt every 1 ms. ISR increments an uint16. This int is returned by Timer_getTick().

Timer_init() can either write directly to the registers or use a struct to write to the registers. To change behavior, change the function call inside Timer_init(). Default: initialise without struct.

UART (Universal Asynchronous Receiver and Transmitter)

Send

There are two options when using UART:

1. Without ISR:

Call uart_init(). Send data with uart_send(). uart_send() is blocking.

2. With ISR:

Call uart_init_isr(). Send data with uart_send_isr(). Data will be stored in a ring-buffer with 512 byte space. For sendig, Transmit Complete Interrupt is used.

Receive

- buf_available:

This function is useful to get information about if there is data in the receive ringbuffer. It returns 1 if there is data available and 0 if not.

- uart_recv: (without ISR, blocking!)

This funciton is used to pull one byte directly from the uart. If there is no byte available, this funciton will be blocking.

- uart_get_data: (with ISR, not blocking!)

If there is data in the receive ringbuffer, one byte will be pulled out and returned. If there is not data, 0 will be returned.

ADC (Analog-Digital-Converter)

There are two analog sources on the ES-Board. Functionality for both will be initialised with adc_init(). The ISR will automatically switch between both sources and retrigger a new adc when the last one is finished.

For each source the ISR will make 10 comnversions before switching to the other source. 8 of them will be stored in an Array. The other two will get dumped.

1. LM35-DZ

The LM35-DZ is a temperatur sensor. it is connected to ADC0 pin. The ADC result is multiplied by 0.4883 to get degree celsius as a result.

adc_get_LM35() will return the average of 8 measured values. (result will be converted to degree celsius)

2. Poti

The Poti will return a value between 0 and 1024.

adc_get_Poti() will return the average of 8 measured values.

Traffic Light

(only Task 2) Traffic Light state machine. There is one traffic light for cars and one for people. Normal state: cars can go, people have to wait. Button1 triggers state change. If the traffic light switched back to cars can go, the button has up to 30 sec delay until trigger will take effect.

Parameter

trafficLight(extraLeds)

Set extraLeds to 0 if you want to use the on-board LEDs. Set extraLeds to 1 if you want to use LEDs connected to PortD (0..4). (Make shure to use Led_init(1) then)

Hardware Setup

- Auto-RED: PD0
- Auto-YELLOW: PD1
- Auto-GREEN: PD2
- Person-RED: PD3
- Person-GREEN: PD4

Software

Call trafficLight() in a loop. There has to be a Timer_getTick() function that returns an uint16 representing millis.

Playground

(Updated so that it uses the timer tick function)

Playground.h provides a playground() function. This function has to be called in a loop.

Functionality

There are four programmes available with this funtion. (One for each button)

The Startup mode is for choosing the program. It is visualised by the led pattern: 10011001. To go back to this mode hold button 1 and 2 for max half a second. If you do so, all leds will turn on, then off and then the indicator pattern will show up.

Program 1

Button test program. Turns on led1 if button1 is pressed. Same for button2, 3 and 4.

Program 2

Board test program. Turns PORTB on and off in intervalls of 500 ms.

Program 3

Lauflicht (chaser light) program.

Program 4

Increment / Decrement program. Press button1 to increment, button2 to decrement.