NTNU

Department of Engineering Cybernetics

Examination paper for TTK4155 Industrial and Embedded Computer Systems Design

Academic contact during examination: Jo Arve Alfredsen

Phone: 90945805

Examination date: Monday 2019-12-09

Examination time (from-to): 09:00 - 13:00

Permitted examination support material: D

Standard pocket calculator permitted.

Printed and handwritten material not permitted.

Other information:

Answers may be given in English or Norwegian

Concise answers are required.

Read the text carefully. Each problem may have several questions.

Exam counts 50% of final grade.

Language: English

Number of pages (front page excluded): 5

Number of pages enclosed: 0

Informasjon om trykking av eksamensoppgave

Originalen er:

1-sidig ☐ **2-sidig** ☐

sort/hvit ☐ **farger** ☐

skal ha flervalgskjema ☐

Checked by:

_____

Date Signature

Problem 1. (50 %)

a. Calculate the minimum and maximum frequencies of the TOV3 interrupt of Timer/Counter 3 of an AVR ATmega162 microcontroller (see Figure 1) running in normal mode at 8MHz.

b. An ultrasonic range sensor measures distance to objects in the range 0.3 – 10 m. The sensor output is made available continuously as a pulse train where the pulse width is proportional to the measured distance, with resolution 1 µs/mm.
Write a driver in C/pseudo-code for the sensor using the microcontroller from a) and the timing mechanisms provided by Timer/Counter 3 to output full resolution distance measurements. The driver should be non-blocking and have the external interface functions:

```
uint8_t sensor_control(uint8_t on) // on=1/0 turn sensor driver on/off
int16_t sensor_getrange_mm(void) // returns distance to object in mm
```

c. What is the purpose of a quadrature encoder (see Figure 3)?
Explain how you can use the quadrature encoder together with the microcontroller from a) to create a rotational position sensor (quadrature counter).
Suggest an implementation of the quadrature encoder and position sensor using C/pseudo-code. The sensor firmware should be able to run in the background as unaffected as possible by other code running on the MCU.

Problem 2. (50 %)

Design case: Suppose that you are hired to design a new room occupancy sensor controller for a building automation company. The sensor should be capable of sensing the presence of people in a room, sensing ambient light, air temperature and quality, controlling room lighting and ventilation, as well as communicating with the building automation central both through wired or wireless network interfaces. The device's detailed specifications are given as follows:

- MCU: The sensor controller should be based on a microcontroller of type AVR ATmega162 as shown in Figure 4. The MCU requires a V_{CC} = 5V power supply and an 8MHz crystal oscillator.
- Occupancy sensor: A pyroelectric passive IR (PIR) sensor with some suitable opamp signal conditioning circuitry is used for this. The output of the sensor is an analog voltage V_{PIR}, where $V_{PIR} > V_{CC}/2$ indicates personnel detection. The comparator inputs of the MCU should be used for this sensor.
- Ambient light sensor: This sensor outputs three analog voltages in the range 0 – 5V which together makes it possible to calculate the room's ambient photopic light level. This will be used to control the intensity of the room lighting fixtures.
- Air temperature and air quality sensors: These two sensors are available as integrated sensor modules each with their own SPI interface.
- Lighting and ventilation control: The room lighting fixtures and ventilation system require two analog input signals in the range 0 – 5V.
- ADC: To read sensor signals, an 8-channel AD converter with 16-bit resolution and a parallel bus interface is available. The ADC should be integrated with the MCU as a pure

memory mapped I/O and requires 32 bytes memory space. The AD converter accepts voltage signals in the range 0 – 5V.

- DAC: To output control signals, a 2-channel DA converter with 8-bit resolution and a parallel bus interface is available. The DAC should be integrated with the MCU as a pure memory mapped I/O and requires 8 bytes memory space. The DA converter generates voltage signals in the range 0 – 5V.
- Wired network interface: A CAN network interface is used as the wired connection to the building automation central. The CAN interface should be based on the SJA1000 CAN controller shown in Figure 2.
- Wireless network interface: A stand-alone wireless communication module featuring both Bluetooth LE 4.2 and IEEE 802.15.4/Zigbee is available. The module integrates the full stack of the two wireless protocols as well as a simple UART for interfacing with the host MCU.
- Two LEDs for status indication should be included.

a. Describe the system in terms of a *high-level* block diagram (a detailed circuit schematic not requested here). Read the specification above in detail, focus on identifying and drawing the individual function blocks/modules that together make up the system, and then indicate the interface between them using simple arrows and labels (no detailed bus/signal connections requested here). If deemed necessary, make your own reasonable assumptions that will make the system work as intended.

b. Make a memory map for the computer with minimal use of address signals, and derive the associated decoding logic (remember that the 1280 lowest addresses (0x0000 - 0x04FF) of the ATmega162 are reserved).

c. The design could be simplified by replacing the function of the DAC with internal MCU resources and some simple external signal conditioning circuitry. How?

d. Draw and explain a circuit schematic that shows how the components of the embedded system should be connected (the details can be limited to central signal lines, but the width of all buses and which ports/bit signals are connected to must be shown). Specify the circuits and signals you find necessary to add.

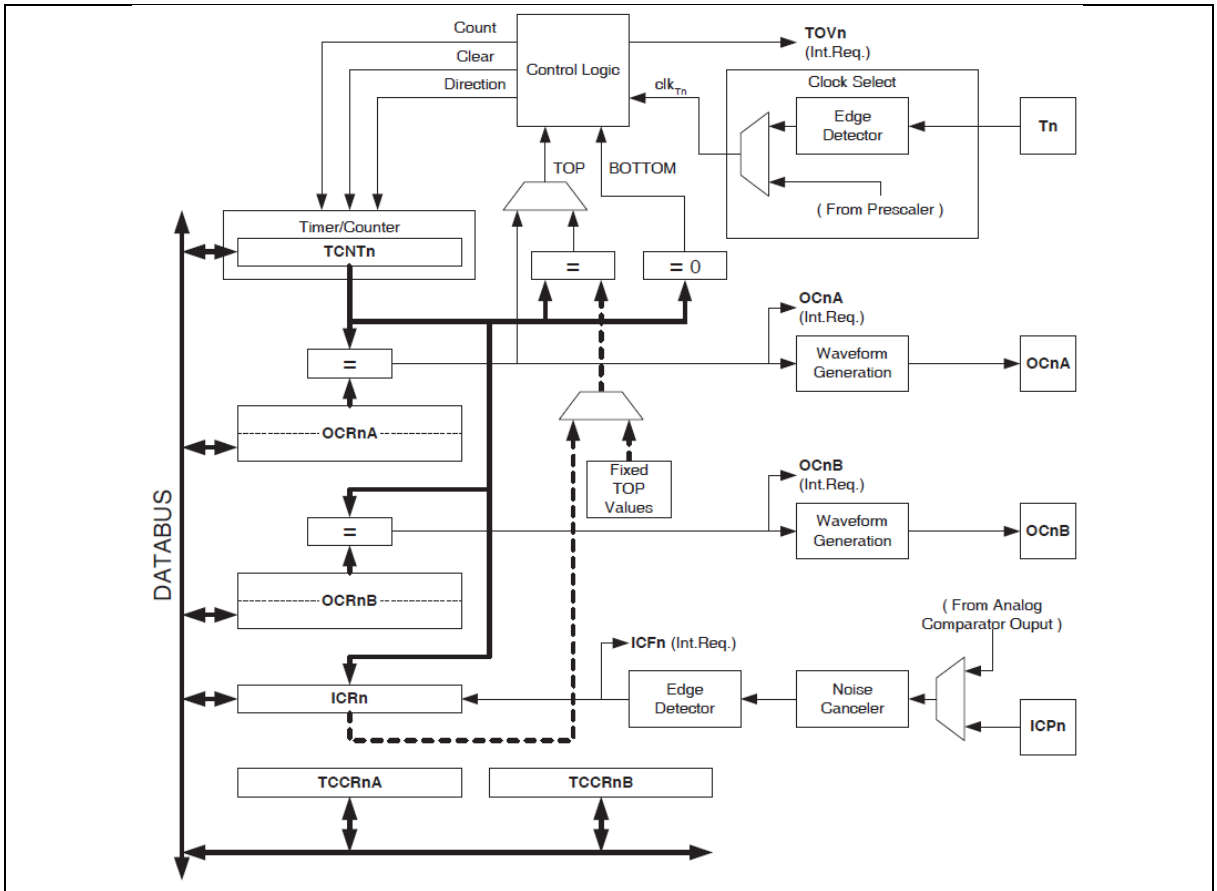

Figure 1. Diagram showing Timer/Counter 1 and 3 in the AVR ATmega162.

- In Timer/Counter n (n = 1 or 3), the TCNTn, OCRnA, OCRnB and ICRn registers are all 16 bit.
- The Timer/Counter can be driven by internal or external clock sources, either directly from the system clock ($clk_{I/O}$), via a prescaled (divided) version of the system clock, or via a clock signal on pin T1 for T/C1. The selectable prescaler values for T/C1 are 8, 64, 256 or 1024. T/C3 has in addition prescaler values 16 and 32, but cannot be clocked by an external source.
- In *Normal* mode, TCNTn increments by one for each clock pulse (0x0000→0x001→…→0xFFFF→0x0000→…). The interrupt signal TOVn (timer overflow) is generated every time the TCNTn overruns from 0xFFFF to 0x0000.
- The ICRn (input capture register) is used in normal mode to capture/timestamp external events in terms of rising and falling edges on the ICPn pin or a trigger signal from the microcontroller's internal Analog Comparator unit. When a rising or falling edge is detected on the ICPn pin or comparator output, the current value of the timer register TCNTn gets loaded immediately into ICRn and the input capture interrupt signal ICFn is generated. The ICESn bit of the timer control register TCCRnB selects which edge on the ICPn pin that is used to capture an event (0 = falling edge, 1 = rising edge).
- The OCRnA and OCRnB are two output compare registers that are used to generate the interrupt signals OCnA and OCnB as well as signal waveforms on the OCnA and OCnB pins when the timer TCNTn matches their content.
- When Timer n is set to *Clear Timer on Compare (CTC) mode*, TCNTn will be cleared to zero when a match with the content of OCRnA is detected and the interrupt signal OCnA will be generated and the OCnA pin will be toggled.
- When Timer n is set to *Fast PWM mode*, TCNTn counts from BOTTOM (0x0000) to TOP and then restarts from BOTTOM again. TOP can be set to fixed values 0x00FF, 0x01FF, 0x03FF or the content of OCRnA or ICRn. The OCnx pin is set on a compare match between TCNTn and OCRnx, and cleared at TOP. A TOVn interrupt is generated when TCNTn rolls over from TOP to BOTTOM.

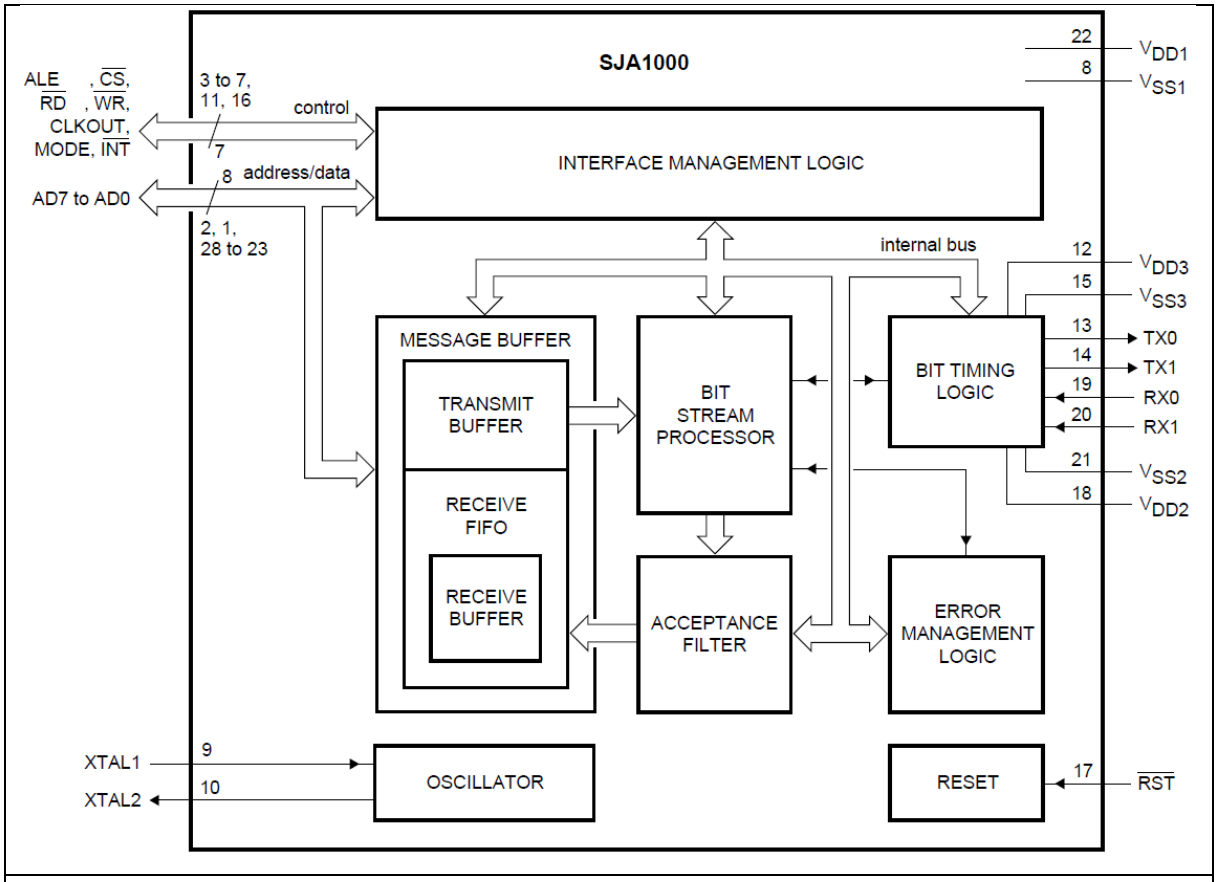

Figure 2. Diagram showing the SJA1000 CAN controller (CAN2.0B).

The SJA1000 features an 8-bit multiplexed address/data parallel bus interface (AD0-7) and a 256 byte internal address space containing CAN receive/transmit buffers as well as status and control registers. With the MODE-pin pulled high, the SJA1000 may be connected directly to the AVR ATmega162 external bus interface.

- ALE, $\overline{RD}$, $\overline{WR}$ and $\overline{CS}$ are control signals controlling bus transactions with the host microcontroller
- $\overline{INT}$ is an active low interrupt signal which is generated by the CAN controller e.g. when a new CAN message is received
- CLKOUT is a clock output signal (not used here)
- $\overline{RST}$ is an active low reset signal
- TX and RX are CAN output and input signals that should be connected to a CAN transceiver
- XTAL1 and XTAL1 are terminals for a 16 MHz crystal
- V_{DD} and V_{SS} are 5 V power and ground, respectively

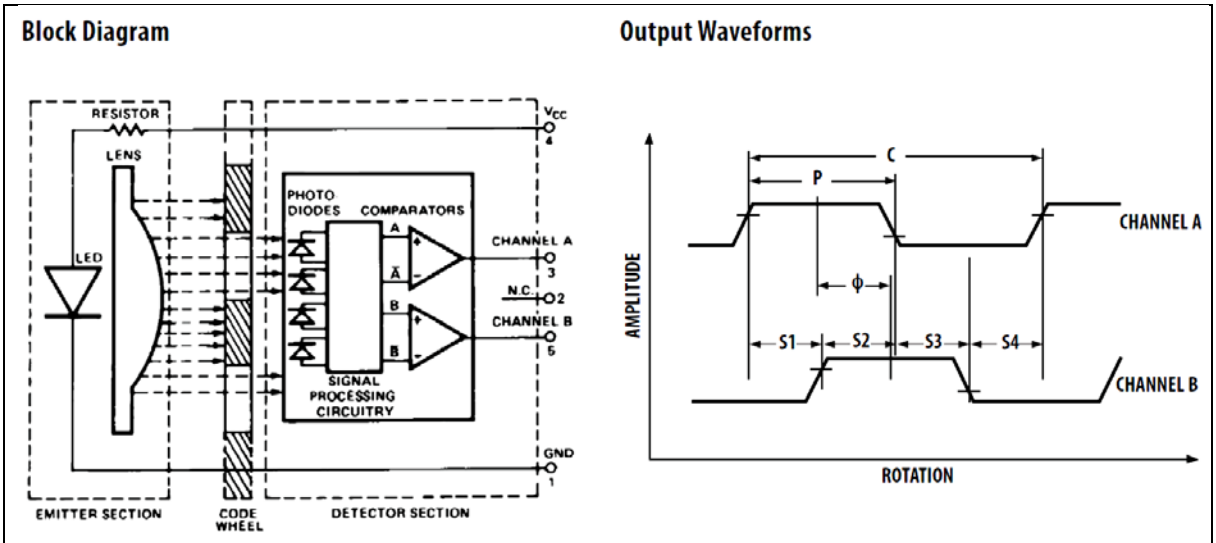

Figure 3. The HEDS-9000 quadrature encoder. The code wheel is divided into N equal slots (bar+hole) and generates N cycles (C) on the output channels A and B for each rotation of the code wheel. N = 500 is a typical value.

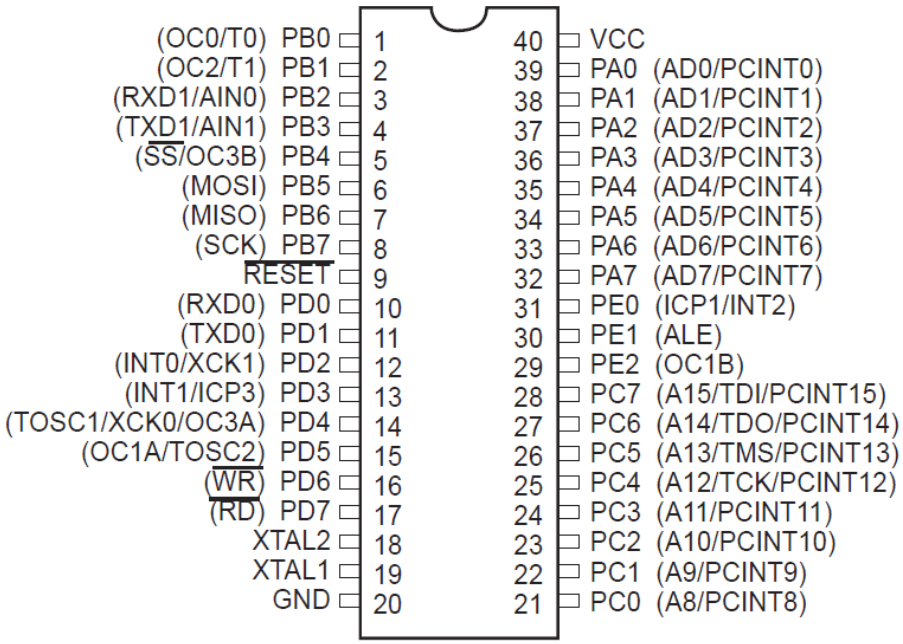

Figure 4. AVR ATmega162.