TECHNISCHE UNIVERSITÄT DRESDEN

cfaed CENTER FOR ADVANCING ELECTRONICS DRESDEN

Networked Embedded Systems Group

Dr. Marco Zimmerling

Networked Embedded Systems WS 2016/17

Sample Solutions to Exercise 2: Communication

Discussion date: December 16, 2016

Task 1: Wired Communication between On-board Components

Figure 1 shows part of the functional block diagram of the Tmote Sky platform. The CC2420 radio chip and the Texas Instruments (TI) MSP430 microcontroller (MCU) communicate through an SPI bus and 6 digital input/output (I/O) lines. The radio reads or sets the state of its I/O pins with frequency f_r, as determined by its crystal oscillator. The MCU reads or sets the state of its I/O pins with frequency f_m, as determined by its internal digitally controlled oscillator (DCO). Assume the two clocks *do not drift* (i.e., f_r and f_m do not change over time), and changes in the state of an I/O pin at one component (i.e., radio or MCU) result in an *instantaneous* change in the state of the respective I/O pin at the other component.

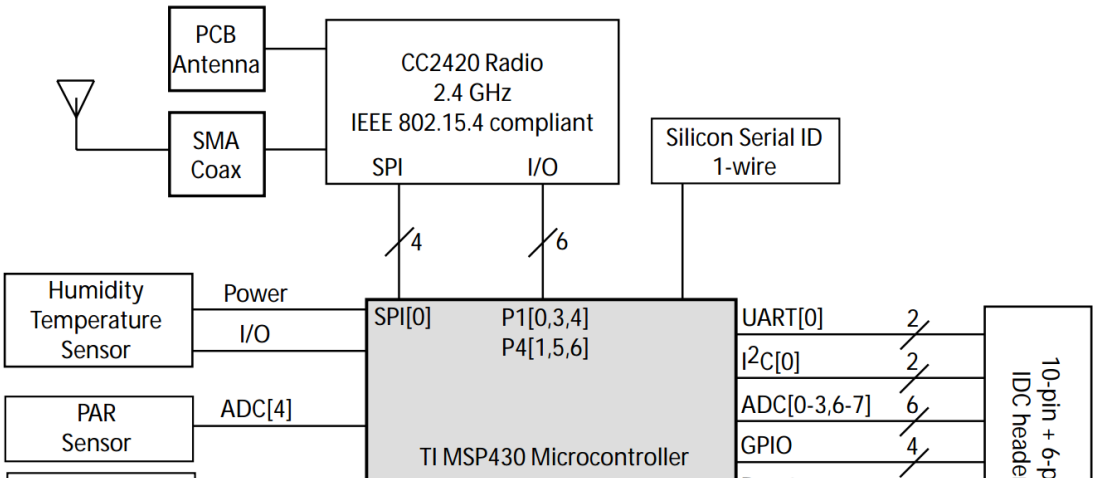

Figure 1: Part of the functional block diagram of the Tmote Sky platform.

Consider the following scenario. The radio signals to the MCU the end of a packet reception by changing the state of a specific I/O pin. As soon as the MCU detects this change, it starts to execute for a fixed number of clock ticks I (with operating frequency f_m). Immediately afterwards the MCU changes the state of a specific I/O pin in order to start a packet transmission by the radio.

(a) Determine the delay D between the end of a packet reception and the start of the next packet transmission assuming that the MCU needs to execute for $I = 100$ clock ticks, and that radio and MCU operate perfectly synchronized with frequency $f = f_r = f_m = 8\,\text{MHz}$.

Solution: This setting is equivalent to the case of a system-on-chip (SoC) platform, where radio and MCU are effectively driven by the same 8 MHz clock source. Thus, as depicted in Figure 2, the delay D between the end of a packet reception and the start of the next packet transmission is simply the time the MCU needs to execute for $I = 100$ clock ticks: $D = I/f = \frac{100}{8{,}000{,}000}\,\text{s} = 12.5\,\mu\text{s}$.

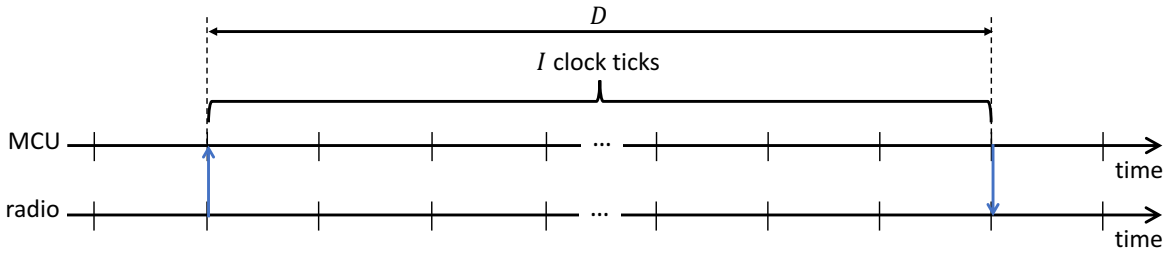

Figure 2: Radio and MCU run perfectly synchronized.

(b) Now assume that radio and MCU run asynchronously with frequency $f = f_r = f_m = 8\,\text{MHz}$. Determine the delay D for the case that the MCU needs to execute for $I = 100$ clock ticks.

Solution: The Tmote Sky is not a SoC platform, so as considered in this subtask the radio and the MCU do not operate in perfect synchrony. Thus, as shown in Figure 3, there is a variable initial delay δ that represents the time it takes for the MCU to detect the signal from the radio, and there is a variable delay $1-\delta$ that represents the time it takes for the radio to detect the signal from the MCU. Because radio and MCU run completely unsynchronized, the initial delay δ is a continuous random variable uniformly distributed in the interval $0 < \delta \leq 1/f$. As a result, compared to the synchronized case, the delay D increases by the duration of one clock period $1/f$: $D = (I+1)/f = \frac{101}{8{,}000{,}000}\,\text{s} = 12.625\,\mu\text{s}$.

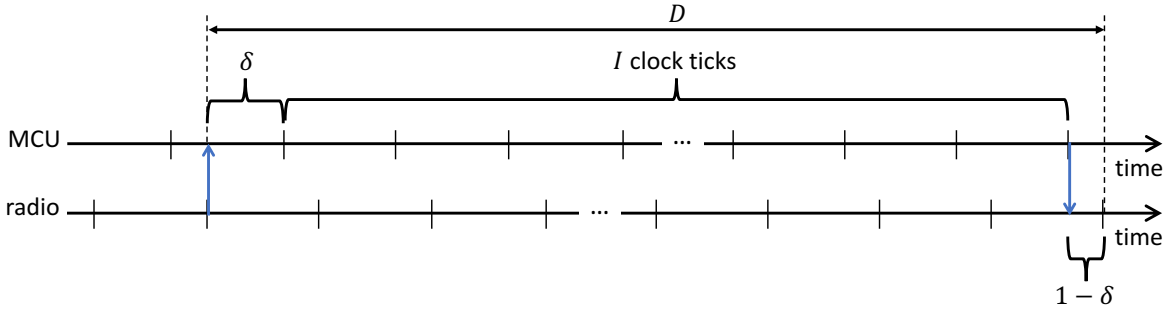

Figure 3: Radio and MCU run completely unsynchronized by with the same frequency.

(c) Now assume that radio and MCU run asynchronously with frequencies $f_r = 8\,\text{MHz}$ and $f_m = 4\,\text{MHz}$. Determine all possible values of the delay D when the MCU executes for $I = 100$ clock ticks. Support your answer by providing an analytical expression for D as a function of f_r and f_m, among others.

Solution: Let us derive an analytical expression for the delay D. To this end, we represent the variable initial delay δ as a fraction k of the MCU clock period $1/f_m$, as shown in Figure 4; that is, k is a continuous random variable uniformly distributed in the interval $0 < k \leq 1$. With that we can express the time it takes for the MCU to detect the signal from the radio and to execute for I clock ticks as

$$(I+k)\frac{1}{f_m} \tag{1}$$

Multiplying (1) by the frequency of the radio clock f_r, we convert this time into the corresponding (fractional) number of radio clock ticks

$$(I+k)\frac{f_r}{f_m} \tag{2}$$

As visible in Figure 4, the radio detects the signal from the MCU only at the next tick of its clock. To account for this, we take the ceiling of (2) to obtain the delay D in terms of radio clock ticks

$$\left\lceil (I+k)\frac{f_r}{f_m} \right\rceil \tag{3}$$

Multiplying (3) by the duration of a single radio clock tick $1/f_r$, we get the delay D in seconds

$$D = \frac{1}{f_r}\left\lceil (I+k)\frac{f_r}{f_m}\right\rceil \tag{4}$$

For $I = 100$, $f_r = 8\,\text{MHz}$, and $f_m = 4\,\text{MHz}$, we have $D = \frac{1}{8{,}000{,}000}(200 + \lceil 2k\rceil)$. Thus, depending on the initial delay represented by k there are two possible values for the delay D:

- $0 < k \leq 0.5 \Rightarrow \lceil 2k\rceil = 1$: $D = \frac{201}{8{,}000{,}000}\,\text{s} = 25.125\,\mu\text{s}$
- $0.5 < k \leq 1 \Rightarrow \lceil 2k\rceil = 2$: $D = \frac{202}{8{,}000{,}000}\,\text{s} = 25.25\,\mu\text{s}$

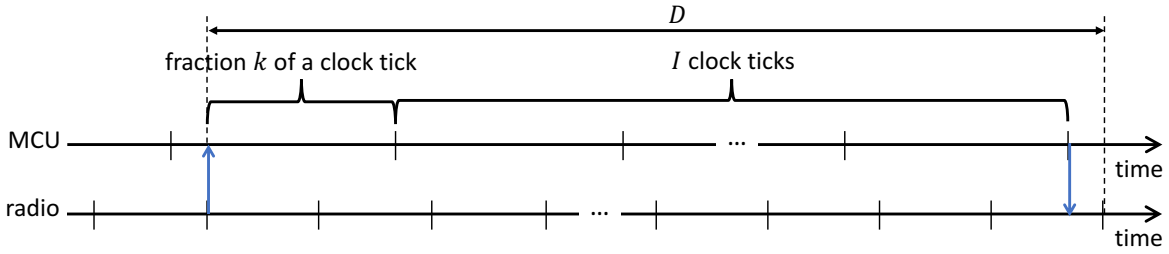

Figure 4: Radio and MCU run completely unsynchronized and with different frequencies.

(d) Assume that radio and MCU run asynchronously with frequencies $f_r = 8\,\text{MHz}$ and $f_m = 2^{22}\,\text{Hz} = 4{,}194{,}304\,\text{Hz}$. How many possible values can the delay D take when the MCU executes for $I = 100$ clock ticks? Is it possible to reduce the number of possible values for the delay D by letting the MCU execute for a few more clock ticks (e.g., by inserting `NOP`s into the code executed by the MCU, where one `NOP` takes exactly one clock period $1/f_m$ to execute)? If so, determine the smallest number of clock ticks greater than 100 for which the number of possible values of the delay D is minimal.

Solution: The delay D can take three possible values for $I = 100$. To see this, we note that since $(I+k)f_r/f_m \approx 190.7 + 1.907 \cdot k$, the smallest possible delay measured in radio clock ticks is $\lceil 190.7\rceil = 191$. We find three ranges for the initial delay represented by k yielding three distinct values for the delay D:

- $0 < k \leq 191 \cdot f_m/f_r - I \approx 0.139 \Rightarrow \lceil (I+k)f_r/f_m\rceil = 191$: $D = \frac{191}{8{,}000{,}000}\,\text{s} = 23.875\,\mu\text{s}$
- $0.139 < k \leq 192 \cdot f_m/f_r - I \approx 0.663 \Rightarrow \lceil (I+k)f_r/f_m\rceil = 192$: $D = \frac{192}{8{,}000{,}000}\,\text{s} = 24\,\mu\text{s}$
- $0.663 < k \leq 1 \Rightarrow \lceil (I+k)f_r/f_m\rceil = 193$: $D = \frac{193}{8{,}000{,}000}\,\text{s} = 24.125\,\mu\text{s}$

Yes, there are certain numbers of clock ticks I for which the delay D takes only two instead of three possible values. The smallest such number of clock ticks greater than 100 is $I = 107$. In this case, the smallest possible delay measured in radio clock ticks is $\lceil 204.08\rceil = 205$. We find

- $0 < k \leq 205 \cdot f_m/f_r - I \approx 0.479 \Rightarrow \lceil (I+k)f_r/f_m\rceil = 205$: $D = \frac{205}{8{,}000{,}000}\,\text{s} = 25.625\,\mu\text{s}$
- $0.479 < k \leq 1 \Rightarrow \lceil (I+k)f_r/f_m\rceil = 206$: $D = \frac{206}{8{,}000{,}000}\,\text{s} = 25.75\,\mu\text{s}$

Using (4) we can determine the delay D for a given I and a set of random values for k uniformly chosen from the interval $]0, 1]$. Figure 5 plots the number of possible values for D depending on I. We can see that for any $I \in \{107, 118, 129, 140, 151, 161, \ldots\}$ there are only two possible values of the delay D. These two values are $1/f_r = 0.125\,\mu\text{s}$ apart.

Task 2: Wireless Communication between Low-power Devices

Two Tmote Sky devices, one sender and one receiver, exchange packets using their IEEE 802.15.4 compliant CC2420 radios. Figure 6 shows the frame format of an IEEE 802.15.4 packet as defined by the standard. The synchronization header (SHR) is automatically generated by the radio hardware. The frame length field

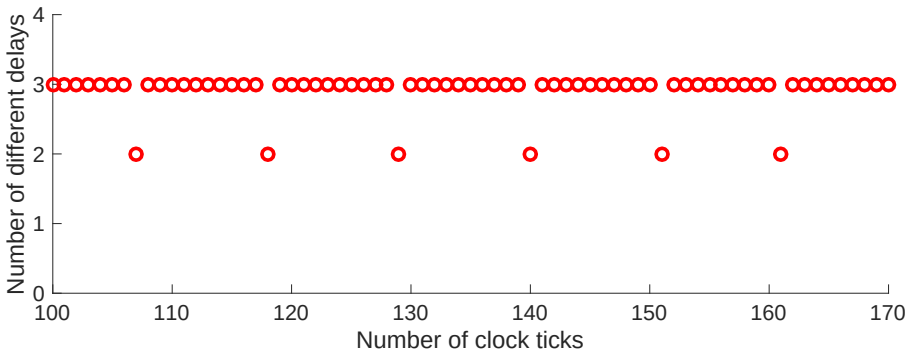

Figure 5: Number of possible values of the delay D depending on the number of clock ticks I the MCU needs to execute, for $f_r = 8\,\text{MHz}$ and $f_m = 2^{22}\,\text{Hz} = 4,194,304\,\text{Hz}$.

(i.e., the PHY header) specifies the number of bytes in the MAC protocol data unit (MPDU); however, the most significant bit of the frame length field is reserved and should be set to zero. Many communication stacks do not use the full MAC layer format. Instead, their MPDU only consists of the MAC payload and the frame check sequence (FCS); that is, the space typically reserved for the MAC header (MHR) can be used to accommodate a larger MAC payload. The FCS is automatically generated by the radio hardware.

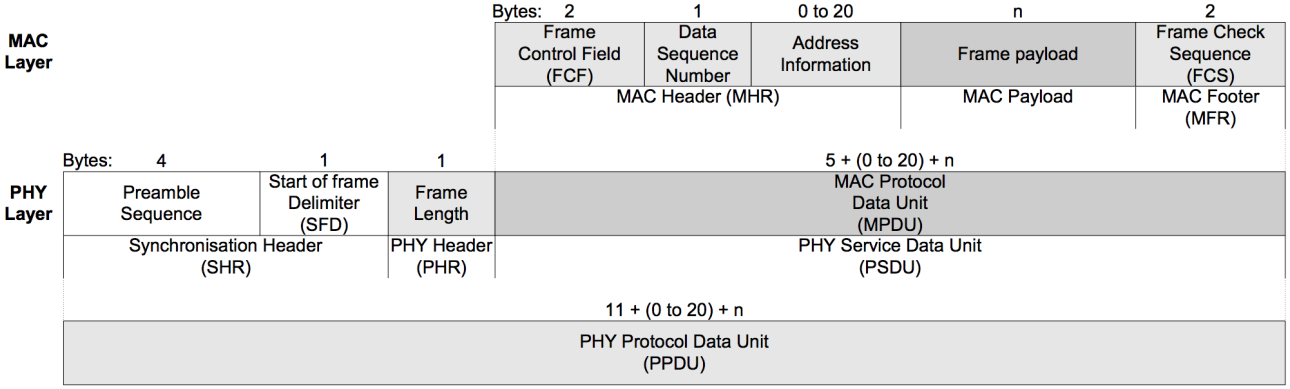

Figure 6: IEEE 802.15.4 frame format.

(a) Determine the maximum size (in bytes) of the PHY protocol data unit (PPDU). What is the maximum size of the MAC payload?

Solution: There are only 7 bits available in the frame length field for specifying the size of the MPDU. Thus, the MPDU can be at most 127 bytes. Subtracting the 2-byte FCS, the maximum size of the MAC payload is 125 bytes. This corresponds to a maximum PPDU of $4 + 1 + 1 + 125 + 2 = 133$ bytes.

(b) The sending device wants to transmit data to the receiving device as fast as possible. The distance between the two devices is small enough so that they can communicate directly with each other. What is the maximum throughput the two devices can theoretically achieve? Keep in mind that the transmit bit rate of a IEEE 802.15.4 radio is $R = 250\,\text{kbit/s}$, and that after triggering a transmission it takes $192\,\mu\text{s}$ until the radio actually starts to transmit the SHR.

Solution: To calculate the maximum single-hop throughput T_s, we note that the MAC payload should be as large as possible, because this minimizes the overhead that comes with each packet (e.g., due to the SHR and the FCS). The maximum MAC payload is 125 bytes. Between two back-to-back transmissions there is an inevitable gap of $192\,\mu\text{s}$, which corresponds to an additional overhead of 6 bytes for a transmit bit rate of $R = 250\,\text{kbit/s}$. Taking into account any other overhead, we get

$$T_s = \frac{125}{4 + 1 + 1 + 125 + 2 + 6} \cdot 250\,\text{kbit/s} \approx 224.8\,\text{kbit/s} \tag{5}$$

(c) Now assume that the distance between sender and receiver exceeds the IEEE 802.15.4 communication range, which is typically on the order of a few tens of meters. Thus, the two devices rely on multi-hop communication, where intermediate devices relay the packets from the sender in a hop-by-hop fashion to the receiver. Assume that the number of hops in the linear multi-hop topology is smaller than the number of IEEE 802.15.4 radio channels (16). Using different channels, two devices can transmit at the same time without interfering with each other. What is the maximum multi-hop throughput the two devices (i.e., sender and receiver that are several hops apart) can theoretically achieve?

Solution: Using multi-hop communication along a path $S, 1, 2, 3, \ldots, R$ that connects sender S with receiver R, there is the problem that, for example, the transmission between node S and node 1 may interfere with the concurrent transmission between node 2 and node 3. To eliminate this so-called intra-path interference, nodes S and 1 and nodes 2 and 3 should use different radio channels. Because using a half-duplex radio a node cannot send and receive at the same time, the maximum multi-hop throughput T_m is half the maximum single-hop throughput, $T_m = T_s/2 \approx 112.4\,\text{kbit/s}$.

(d) Assume that sender and receiver are $H = 6$ hops apart; that is, 5 intermediate devices relay packets from the sender to the receiver. The wireless channel conditions are difficult: single-hop transmissions between any two devices succeed only with probability $p = 0.5$. To still achieve a high end-to-end reliability, devices can re-transmit each packet up to N times. A device re-transmits a packet if it does not receive an acknowledgment within a certain interval after a (re-)transmission. How many per-hop retransmissions N are needed to achieve an average end-to-end reliability higher than 99 %?

Hint: You may assume that packet (re-)transmissions are statistically independent events.

Solution: Let as derive an expression for the end-to-end reliability R along a H-hop path, where each link along the path has exactly the same probability of success p for a single transmission. We can write

$$R = p_{hop}^H \tag{6}$$

where p_{hop} denotes the probability of success across any given link after N retransmissions. The probability that the packet is *not* successfully received across a link after the initial transmission and N subsequent re-transmissions is $(1-p)^{N+1}$. Thus, we have $p_{hop} = 1 - (1-p)^{N+1}$ and hence

$$R = [1 - (1-p)^{N+1}]^H \tag{7}$$

For $p = 0.5$ and $H = 6$ we need to allow for $N \geq 9$ packet re-transmissions at each hop to achieve an average end-to-end reliability of $R > 0.99$.

(e) To achieve a lifetime of multiple years, a device should duty-cycle its radio. Assume a device runs a sender-initiated media access control (MAC) protocol based on low-power listening (LPL). The MAC protocol is configured to regularly wake-up the radio every T_w for $T_{on} = 10\,\text{ms}$ to check whether there is any incoming traffic. When the radio is turned on it draws $I_{on} = 18.8\,\text{mA}$, and $I_{\mathit{off}} = 20\,\mu\text{A}$ when it is turned off. How long must the wake-up interval T_w be to achieve an estimated lifetime of 2 years (i.e., 730 days), assuming the device is powered by batteries that supply 2000 mAh at 3 V. Think about the implications on the achievable throughput.

Hint: Assume the device never sends or receives anything, it does nothing else than duty-cycling its radio. Also, neglect any effects related to battery discharge.

Solution: Given the battery capacity, Q, and the current draws when the radio is on, I_{on}, and off, I_{off}, we can determine the lifetime T of a device that only duty cycles its radio as

$$T = \frac{Q}{D_{on} I_{on} + D_{\mathit{off}} I_{\mathit{off}}} \tag{8}$$

where D_{on} and D_{off} are the fractions of time the radio is on and off. The latter two can be expressed as $D_{on} = T_{on}/T_w$ and $D_{\mathit{off}} = 1 - D_{on} = 1 - T_{on}/T_w$. Substituting into (8) and solving for T_w gives

$$T_w = \frac{Q/T - I_{\mathit{off}}}{T_{on}(I_{on} - I_{\mathit{off}})} \tag{9}$$

For $Q = 2000\,\text{mAh} = 7200\,\text{As}$, $T = 2 \cdot 365 \cdot 24 \cdot 60 \cdot 60\,\text{s}$, and T_{on}, I_{on}, and I_{off} as given above, we find that the wake-up interval T_w must be $1.994578\,\text{s} \approx 2\,\text{s}$ or longer to achieve an estimated lifetime of 2 years.

The throughput in such a low-duty-cycle network is significantly lower than the theoretical maximum. Without specific MAC protocol optimizations, a device is only able to receive one packet every two seconds, and a sender needs to wait on average one second until the intended receiver wakes up. As a result, only applications with low traffic load can be supported, such as environmental monitoring.