CIS 441 & 541: Embedded Software for Life-Critical Applications

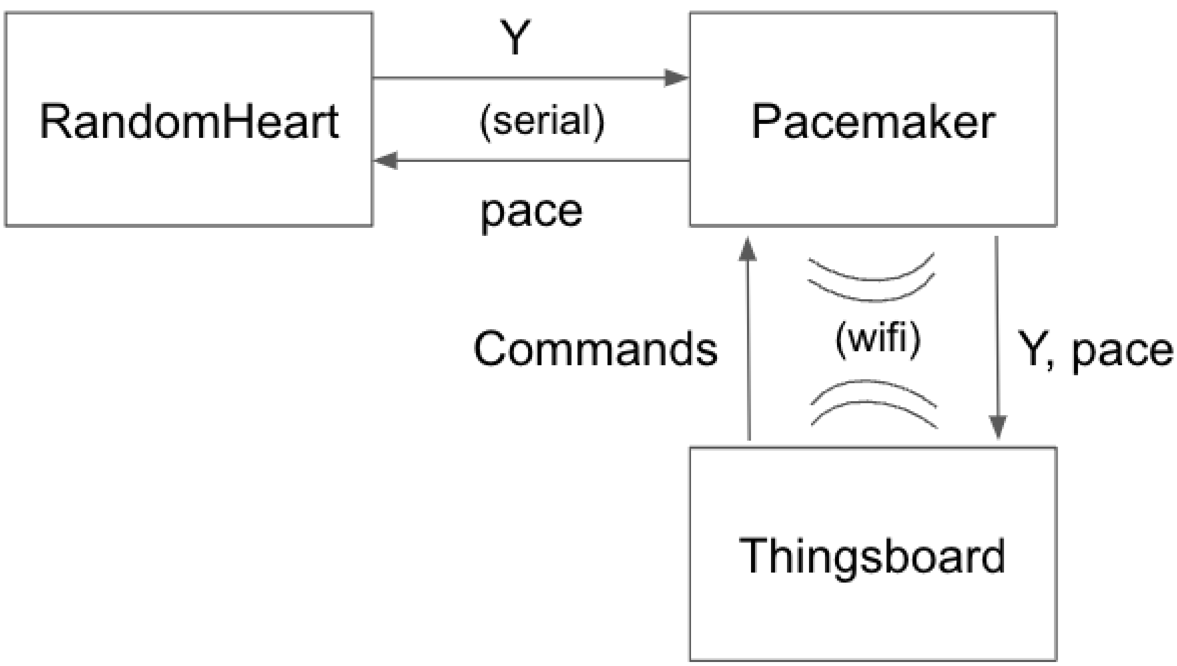
Cloud-based Smart Pacemaker-Challenge Project

Milestone 1: Implementing Heart and Pacemaker on MBED, Connecting to the Cloud with MQTT

Due:
Milestone 1(a) 3:30 pm, 2/28, 20% of project
Milestone 1(b): 3:30 pm, 3/16, 10% of project

In this milestone, you have to implement a RandomHeart and a Pacemaker.

At a high level, a RandomHeart will read a pace value (either 0 or 1) and output the amplitude of the heart's ECG signal (Y). The corresponding Pacemaker will read in the amplitude, improve its understanding of the ECG signal over time and output a pace value.



There are three parts as described below

(a) Heart and Pacemaker - 20% of project

Create a RandomHeart program that runs on one MBED and a Pacemaker program that runs on another MBED. The two programs should communicate by writing and reading from a shared port. The instructions for wiring the MBEDs together are provided in Appendix 2.

For upcoming milestones, it would be helpful for you to think of properties that you believe are necessary to establish the correctness of your design.

(b) Thingsboard (MQTT/wifi) + Remote Control - 10% of project

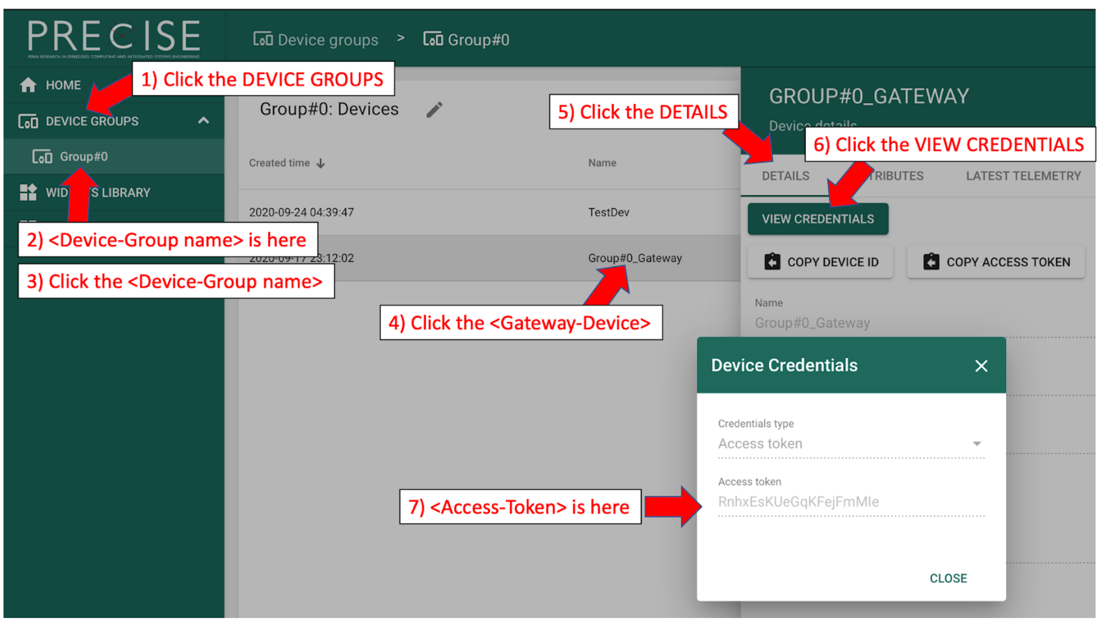
Now that you have your heart and pacemakers running, please create a new thread on the pacemaker MBED to communicate with Thingsboard via MQTT. This thread should parallel to the thread in part (a) on the pacemaker.

The thread will send out the Y and pace values via MQTT to Thingsboard. It will also take input commands to start, stop and update pacemaker parameters (LRL, URL, HRL, VRP) and execute these commands.

Prerequisites and Preparations:

1. ThingsBoard account. Once you have your team, email aawatson@seas.upenn.edu to be added to your team dashboard. Each member of the project group will be given an individual account.

2. The name of Device-Group and Access-Token.
   To communicate with the ThingsBoard using MQTT, two key information from the ThingsBoard is required for each project group. This information can be obtained from the ThingsBoard with the following steps:

   For more detailed information, please refer to the ThingsBoard tutorial slides.

3. Dashboard.
   It's the main component for the visualization like the same as you have seen in the HW4. Initial dashboard will be provided by default. For creating a new dashboard, please refer to the ThingsBoard tutorial slides.

MQTT topic and data format summary

There are some changes in the MQTT information from the HW4. The changes are highlighted as bold and italic. It is noted that the information surrounded by “<” and “>” must be replaced with the actual value corresponding to the information.

1. ThingsBoard MQTT Broker Information: The Access-Token must be used in the ‘Username’.
    - Broker address: 158.130.55.5:8883
    - Username: ***<Access-Token>***
    - TLS must be enabled.
    - Password is not required.
    - MQTT QoS: 1
2. Topic - v1/gateway/connect: Your own pacemaker name should be used in the ‘device’ field and the Device-Group name must be used in the ‘type’ field in the JSON string.

```
{
    “device”: “<pacemaker name>”,
    “type”: “<Device-Group name>”
}
```

3. Topic - v1/gateway/attributes: You can send either a single (y, pace) tuple or multiple (Y, pace) values in a single message and additional parameters/values can be included in the pacemaker attribute if needed.

```
{
  “pacemaker”: {
       “Y”: <Y value>,
       “pace”: <pace value>
}
```

Known bugs and tips

1. Unless you refresh the Dashboard, newly created devices are not shown in the Dashboard.

Notes

Please refer to the ThingsBoard tutorial slides and video for the detailed usage of the ThingsBoard.

Evaluation Criteria: Please submit all of your code in a zip file with a detailed README to Canvas by the deadline given above. **Attach a screen capture video of the real-time plot of Y and pace values on thingsboard to your submission.**

Extra Credit:

1. Implementing a more comprehensive mode such as VVIR (See Table in Appendix 1a) will give you extra-credit and will be graded in an in-person demo with a TA/grader.
2. Splitting the functionality of the pacemaker in part (a) into two threads (total 3 threads on the pacemaker) where one thread communicates with the heart and another thread computes pace values will also give extra-credit. Document this in your submission README and your final report.

Extra credit will be tallied separately and could be used to increase your final letter grade if your current total score is just below the cutoff for the next higher letter grade. Extra credit will only benefit you and not change the grading curve for the rest of the class.

Late policy: 20% off for each day

References

[1] Boston Scientific. "Pacemaker System Specification." Boston Scientific (2007). http://sqrl.mcmaster.ca/_SQRLDocuments/PACEMAKER.pdf
[2] MBED thread: https://os.mbed.com/docs/mbed-os/v6.15/apis/thread.html
[3] MBED mutex: https://os.mbed.com/docs/mbed-os/v6.15/apis/mutex.html
[4] Additional information: https://thoracickey.com/6-pacemaker-timing-cycles-and-special-features/

Appendix 1a: Detailed description of the working of a VVI pacemaker (and the corresponding RandomHeart)

The below Table (Table 6 in [1]) summarizes each characteristic of a VVI pacemaker and that of pacemakers in other modes. Additional information for different modes, LRL, URL, VRP (Ventricular Refractory period) can be found in [1, 4]. **We enumerate the important implementation aspects below in points (a)-(c).**
(For the moment, "Rate Smoothing" is not required.)

Parameter	AAT	VVT	AOO	AAI	VOO	VVI	VDD	DOO	DDI	DDD	AOOR	AAIR	VOOR	VVIR	VDDR	DOOR	DDIR	DDDR
Lower Rate Limit	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Upper Rate Limit	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Maximum Sensor Rate											X	X	X	X	X	X	X	X
Fixed AV Delay							X	X	X	X					X	X	X	X
Dynamic AV Delay							X			X					X			X
Sensed AV Delay Offset										X								X
Atrial Amplitude	X		X	X				X	X	X	X	X				X	X	X

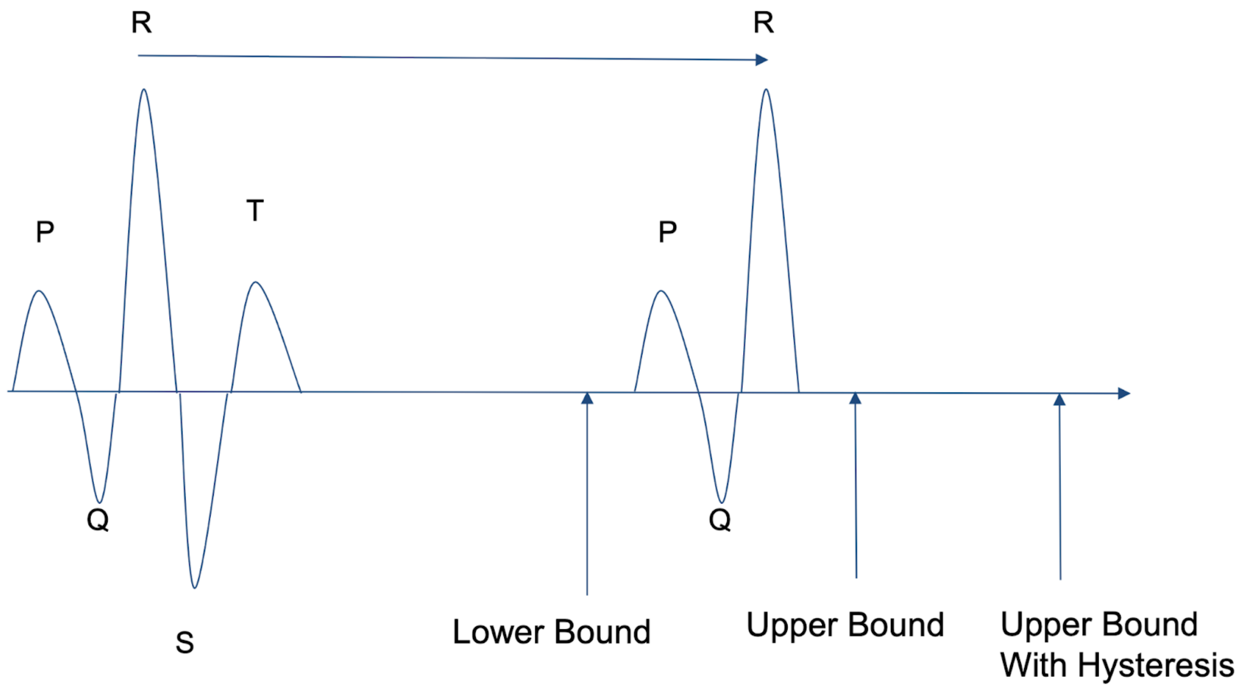
Ventricular Amplitude		X			X	X	X	X	X	X			X	X	X	X	X	X
Atrial Pulse Width	X		X	X				X	X	X	X	X				X	X	X
Ventricular Pulse Width		X			X	X	X	X	X	X			X	X	X	X	X	X
Atrial Sensitivity	X			X					X	X		X					X	X
Ventricular Sensitivity		X				X	X		X	X				X	X		X	X
VRP		X				X	X		X	X				X	X		X	X
ARP	X			X					X	X		X					X	X
PVARP	X			X					X	X		X					X	X
PVARP Extension							X			X					X			X
Hysteresis				X		X				X		X		X				X
Rate Smoothing				X		X	X			X		X		X	X			X
ATR Duration							X			X					X			X
ATR Fallback Mode							X			X					X			X
ATR Fallback Time							X			X					X			X
Activity Threshold											X	X	X	X	X	X	X	X
Reaction Time											X	X	X	X	X	X	X	X
Response Factor											X	X	X	X	X	X	X	X
Recovery Time											X	X	X	X	X	X	X	X

(a) An ECG wave, **normally**, is composed of many wavelets, each with P, Q, R, S, T sinusoidal waves with different amplitudes and frequencies. For a normal heart, no pacing is required (*i.e.* pace = 0).
(b) The R-R separation between consecutive wavelets in a normal heart has a lower and upper bound. The lower bound is given by the inverse of URL and upper bound is given by the inverse of LRL.

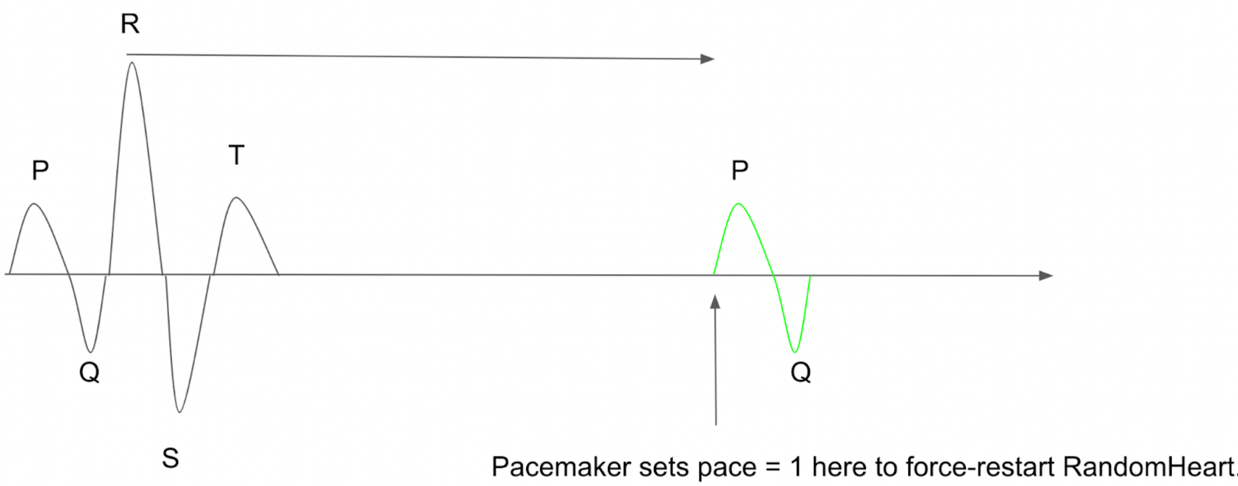
Example 1: If the LRL = 40 beats / min and the URL = 60 beats / min, then

- the upper bound = 1 / LRL = 1 / 40 bpm = 1.5 seconds / beat
- the lower bound = 1/ URL = 1 / 60 bpm = 1 seconds / beat

In this example, the next R wave should be between 1 second and 1.5 seconds of the previous R wave.

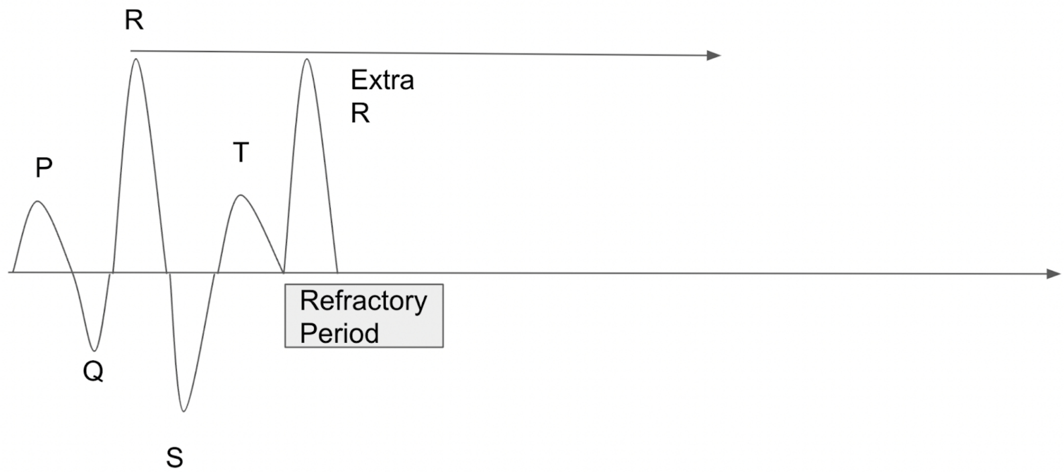


(i) In case the RandomHeart doesn't beat in time for the R-R interval to lie within its upper bound, the pacemaker needs to send a pace=1 signal at the appropriate time. (see below figure)



(ii) In case, the RandomHeart beats early such that the R-R interval is less than its lower bound, the pacemaker needs to start measuring the R-R interval from the new R wave and maintain pace=0.

(iii) Alternatively, in case of an anomalous additional R wave after PQRST that lies within the **refractory period,** the pacemaker must ignore the anomalous R wave. (see below figure)



(iv) Please note that it is possible to have relevant combinations of the above cases in the output of RandomHeart. Create your own comprehensive RandomHeart to test your pacemaker code adequately!

(c) In order to promote self pacing (where the heart paces itself, without the help of the pacemaker), the pacemaker controller should include hysteresis pacing.
   (i) When a natural R-wave is sensed (excluding during the refractory period), hysteresis pacing is enabled. When hysteresis pacing is enabled, the pacemaker allows for a longer interval after sensing an R wave to pace the heart. In other words, the lower rate limit (LRL) is lowered to the hysteresis rate limit (HRL).
   (ii) Otherwise, when a paced R-wave is sensed, the hysteresis pacing is disabled and LRL is left at the original value.

<u>Example 2:</u> In Example 1, if HRL = 30 beats / min, then

- If a natural R wave is sensed, LRL is set to HRL. The upper bound is now equal to 1/ 30 bpm = 2 seconds / beat. This means that the next R wave should be between 1 second and 2 seconds of the previous R wave.
- Otherwise, if a natural R wave is not sensed, the LRL is not changed. Just like in Example 1, the next R wave should be between 1 second and 1.5 seconds of the previous R wave.

<u>In your reports, add the following table with all your choices of these variables,</u>
Please note that the heart execution frequency should be greater than the pacemaker execution frequency.

Heart Execution Frequency (Hz)	
Pacemaker Execution Frequency (Hz)	
PQRST wave amplitudes (Volts) and time intervals (seconds)	P wave: Q wave:

	R wave: S wave: T wave:
URL (beats per minute)	
LRL (beats per minute)	
Refractory Period (seconds)	
HRL (beats per minute)	

An example of the above table is given below

Heart Execution Frequency (Hz)	100 Hz
Pacemaker Execution Frequency (Hz)	50 Hz
PQRST wave time intervals (seconds), amplitudes (milli Volts)	P wave: 0.11 seconds, 0.2 mV Q wave: 0.02 seconds, - 0.2 mV R wave: 0.07 seconds, 0.5 mV S wave: 0.02 seconds, - 0.2 mV T wave: 0.3 seconds, 0.2 mV
URL (beats per minute)	60 bpm
LRL (beats per minute)	40 bpm
Refractory Period (seconds)	0.5 seconds
HRL (beats per minute)	30 bpm

Appendix 2: Instructions to wire the MBEDs together and communicate over shared port

Heart MBED	Pacemaker MBED
Pin 9 - TX	Pin 10 - RX
Pin 10 - RX	Pin 9 - TX
Vin	VU
GND	GND

Note: Make sure your Heart and Pacemaker are connect to the same GND

You can write to the serial port as given in the below screenshot:

```
#include "mbed.h"

Serial pacemaker(p9, p10);

int main(int argc, char* argv[]) {
    pacemaker.baud(9600);
    int i=0;
    while(1){

        pacemaker.printf("Pacemaker: %d", i);

        i = i+1;

        wait(1);
    }
}
```