

The Pacemaker Project

Amanda Watson

Department of Computer and Information Science

School of Engineering and Applied Science

University of Pennsylvania

CIS 441/541 : Embedded Systems for CPS/IoT, Spring 2022

Contents

- I. Introduction to Pacemaker Challenge
- II. Background Information
- III. Pacemaker System
- IV. Pacemaker Timing Cycles
- V. Project Overview

The Pacemaker Challenge

- The formal method challenge problem issued by the **Software Certification Consortium** (SCC), hosted by the McMaster University's **Software Quality Research Lab** (SQRL)
- **Boston Scientific** has released into the public domain the system specification for a previous generation pacemaker
 - This offers an opportunity for the formal methods community to propose novel ideas for pacemaker design
- **The reference hardware platform** was developed by a ECE design group at the University of Minnesota.

The Pacemaker Challenge

- Requirements Document
 - 35 pages, informal natural language (English) and tables
- Main areas:
 - System: DCM, leads, pacing pulse, brady modes and state.
 - Diagnostics: monitoring, battery status, etc.
 - Bradycardia therapy: definition of user programmable parameters (e.g., rate limits, delays).
- Domain knowledge has been summarized in a book
 - S. Serge Barold, Roland X. Stroobandt, Alfons F. Sinnaeve, *Cardiac Pacemakers Step by Step: An Illustrated Guide* (2004)
- SQRL Websites
 - <http://www.cas.mcmaster.ca/sqlr/pacemaker.htm>

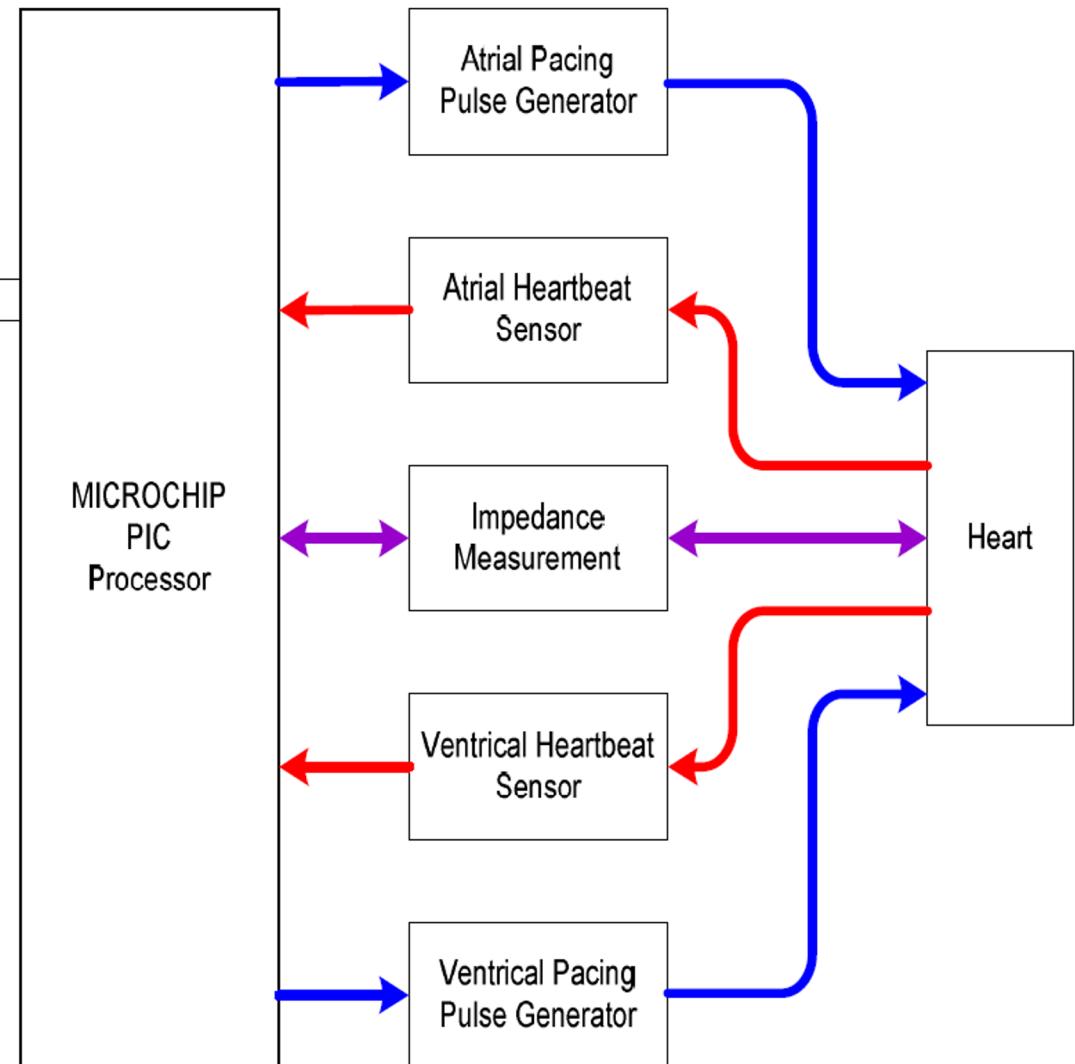
The Pacemaker System Specification

- Defines functions and operating characteristics of the pacemaker system
- Identifies the system environmental performance parameters of the system
- Characterizes anticipated uses of the system
- Includes
 - System definition
 - System requirements
 - Diagnostics information
 - Bradycardia therapy information

The Pacemaker Challenge—Objectives

- Closing the gap between software modeling and code
 - Formal model for the operation of a cardiac pacemaker under useful operating mode. (e.g., VVI and DDD mode)
 - Implementation for the pacemaker device that adheres to the formal models with proper verifications on PIC board.
- The challenge has multiple dimensions
 - Participants may choose to submit a complete version of the pacemaker software
 - may choose to submit just a formal requirements documents - or anything in-between

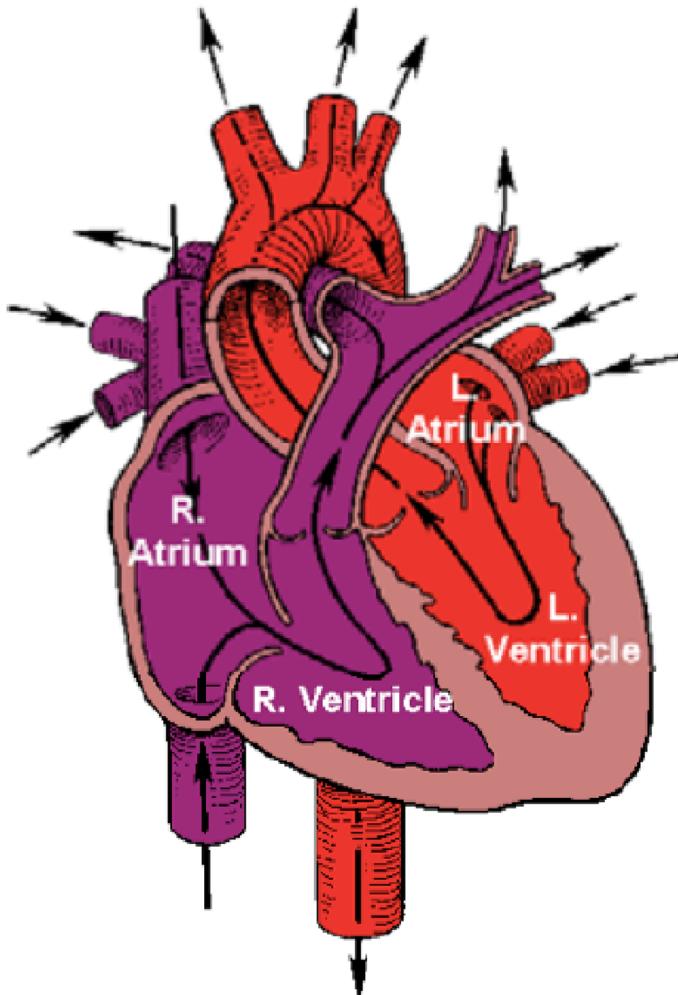
The Pacemaker Challenge—Target Board



Contents

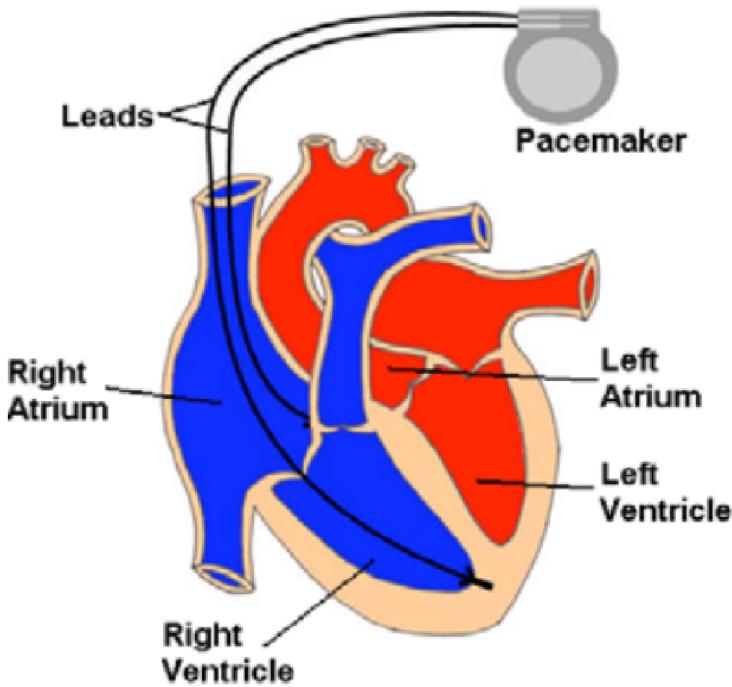
- I. Introduction to Pacemaker Challenge
- II. Background Information
- III. Pacemaker System
- IV. Pacemaker Timing Cycles
- V. Project Overview

Background: The human heart



- Four chambers: atria & ventricles
- Electrical stimulus in the right atrium → heart's chambers contract & pump blood → the ventricles do the same
- When this system does not work properly, a pacemaker may be used to regulate the heart rate

Cardiac Pacemaker

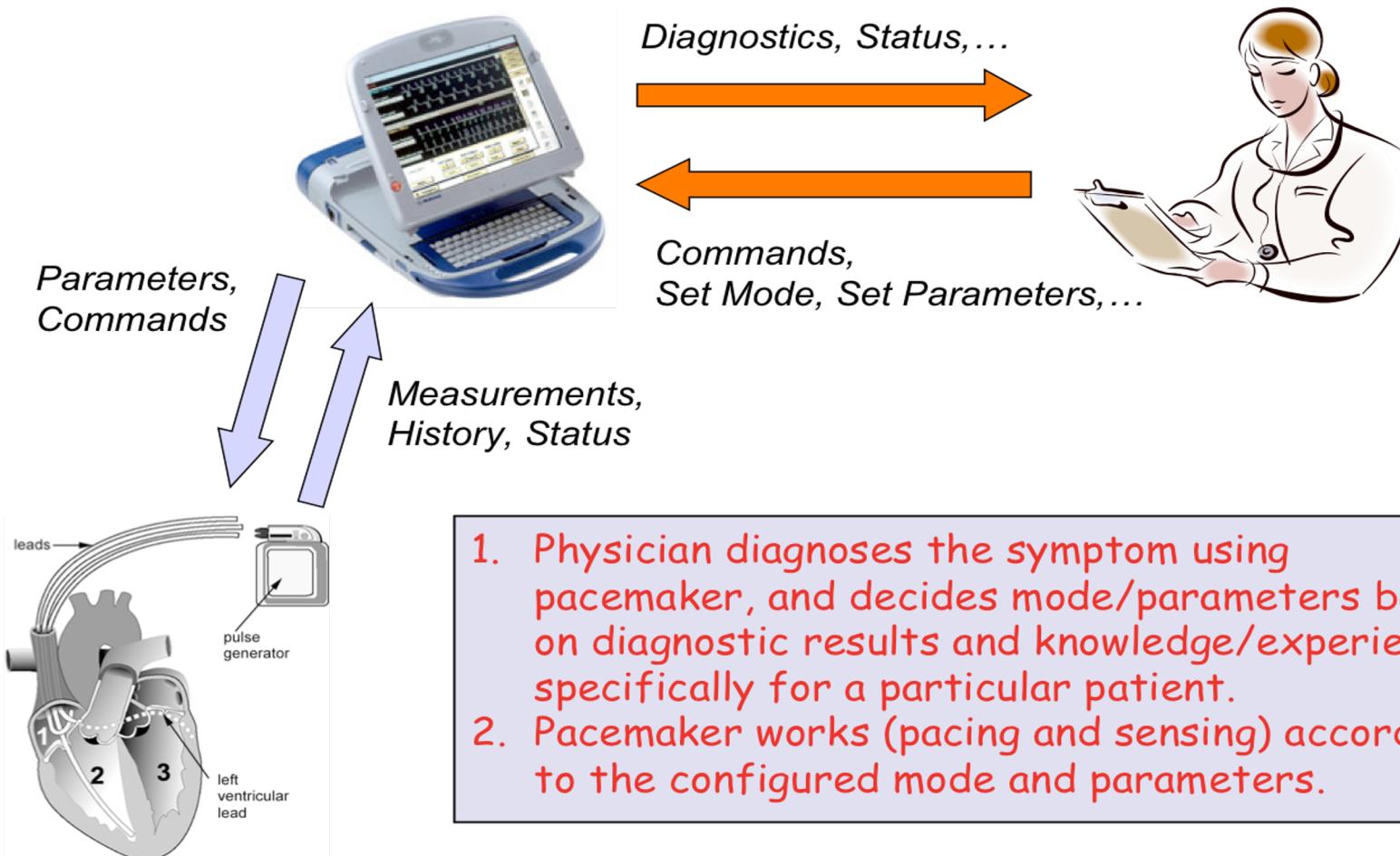


- Deliver electrical stimuli, or **paces**, over leads with electrodes that are in contact with the heart
- May detect natural cardiac stimulations, called **senses**
- Must satisfy **requirements**:
 - The heart rate must not be too low
 - The ventricles must contract at a particular interval after the atria contract

Contents

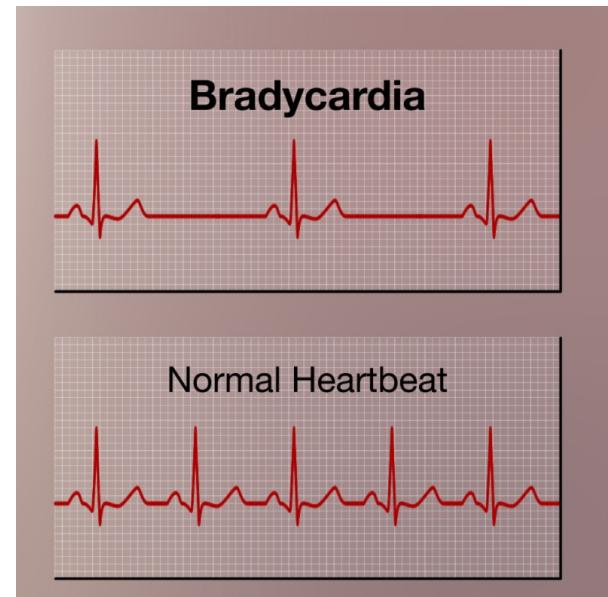
- I. Introduction to Pacemaker Challenge
- II. Background Information
- III. Pacemaker System
 - i. Components
 - ii. Operating Modes
 - iii. Parameters
- IV. Pacemaker Timing Cycles
- V. Project Overview

Pacemaker Usage



System Components for Pulse Generator Device

- Detects and provides therapy for bradycardia conditions
- Provides programmable, single- and dual-chamber, rate-adaptive pacing, both permanent and temporary
- In adaptive rate modes, an accelerometer is used to measure physical activity resulting in a sensor indicated rate for pacing the heart
- Provides sensor output data and rate histograms
- After implantation, interacts and provides diagnostic features including
 - Real-time telemetry markers
 - EGMs
 - P and R wave measurements
 - Lead impedance
 - Battery status tests



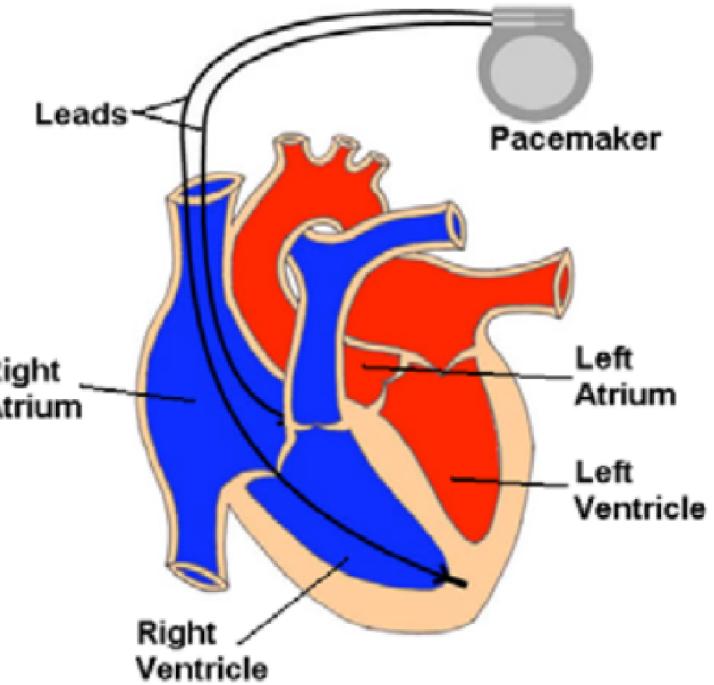
System Components: Device Controller-Monitor

- Used for primary implantation, pre-discharge electrophysiology (EP) support, and follow-up support for the pacemaker system
- Programs and interrogates the device
- Commands delivery of a “Pace Now” pace
- Acquires and shows diagnostics and lead signal measurement information, sensor history and trending information, and multi-channel monitoring
- Monitors battery status



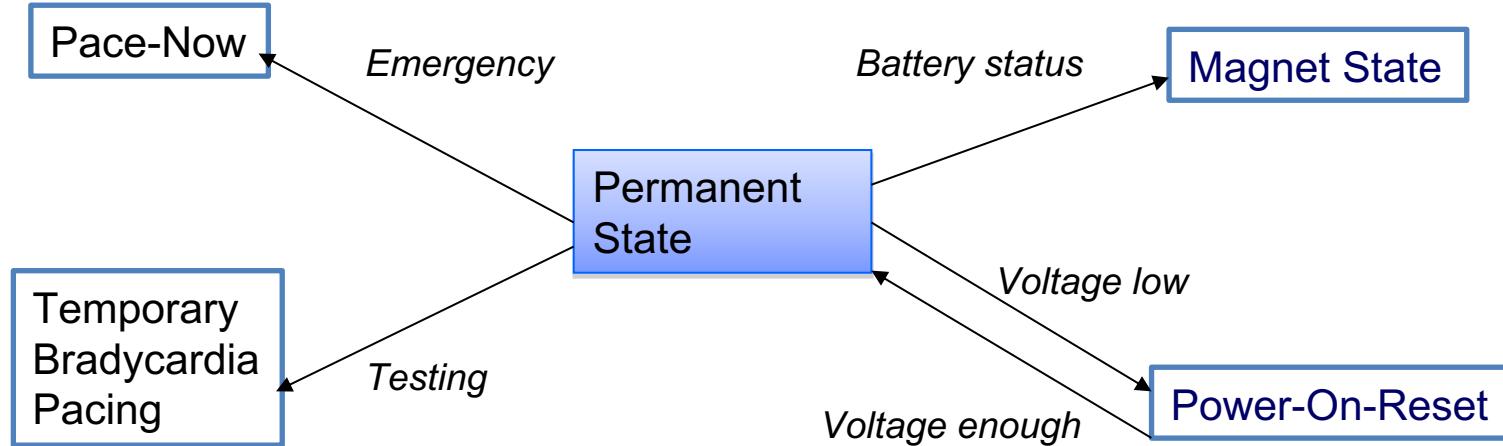
System Components: Lead System

- Implanted in the patient
- Allows the device to sense intrinsic activity of the heart's electrical signals
- Delivers pacing therapy to the patient's heart
- Leads are connected to the pulse generator via its header



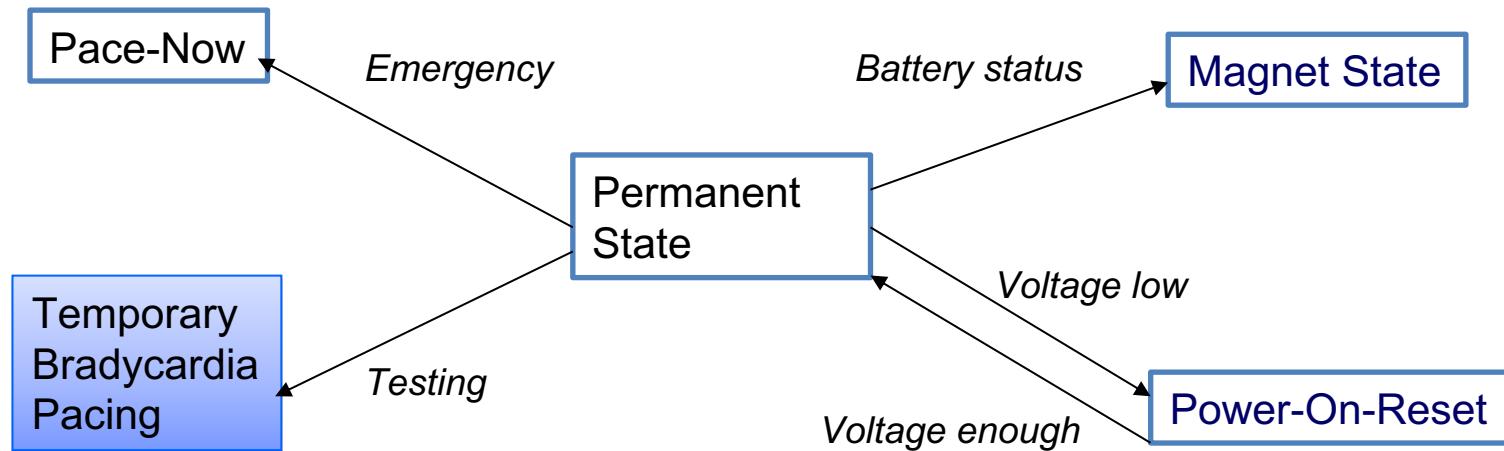
Pacemaker Operating States

Permanent State: Normal state of operation with programmed parameters



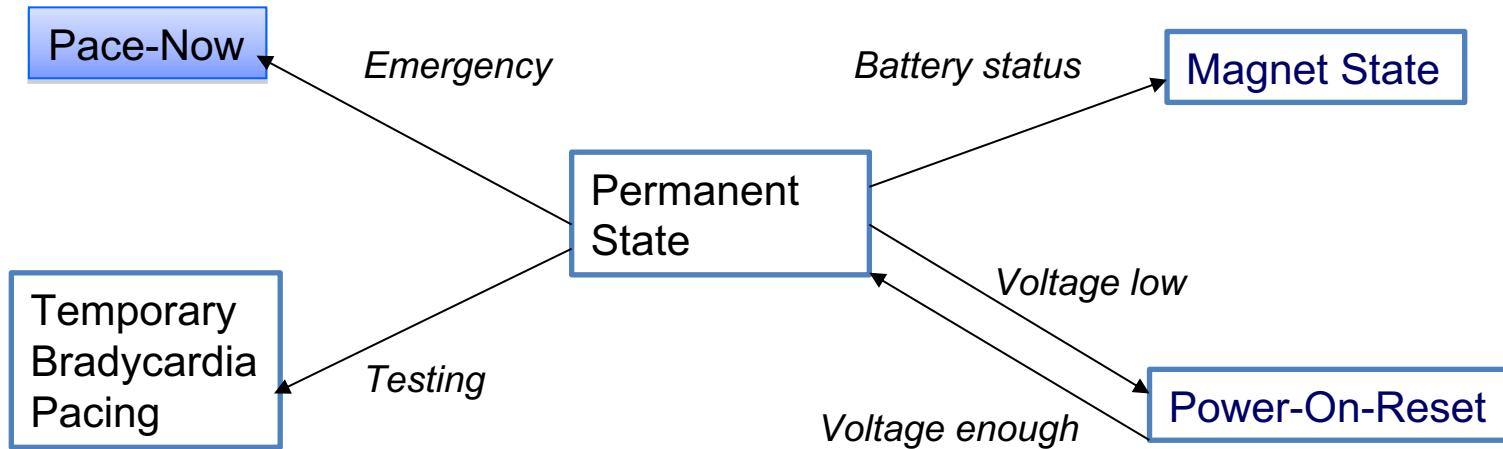
Pacemaker Operating States

Temporary Bradycardia Pacing: Temporarily test various system parameters or provide patient diagnostic testing



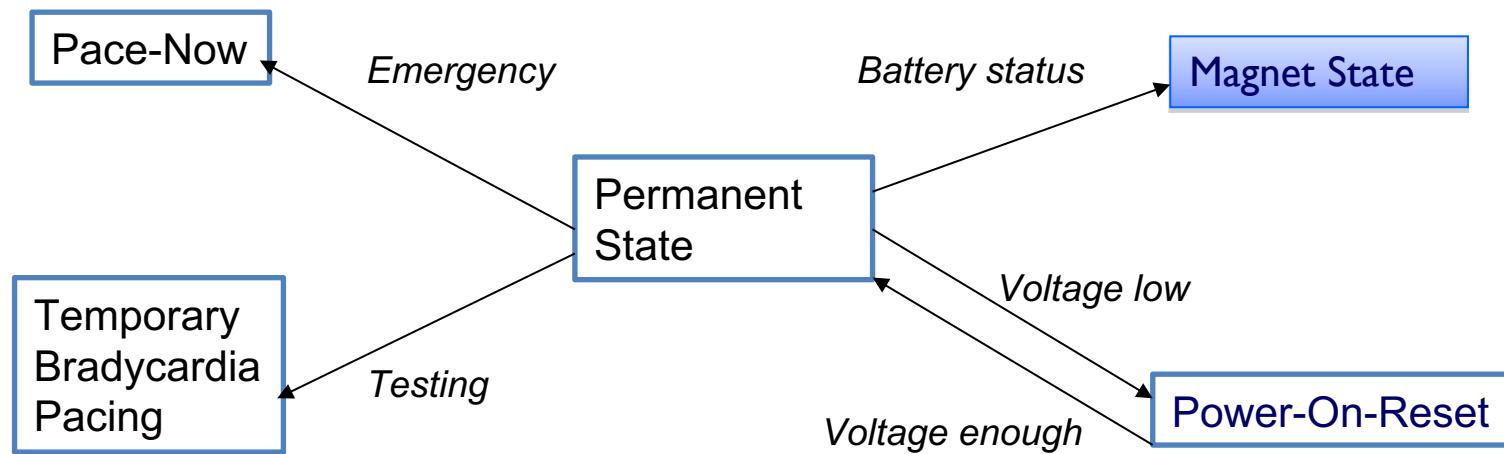
Pacemaker Operating States

Pace-Now : Emergency pacing, with fixed parameters



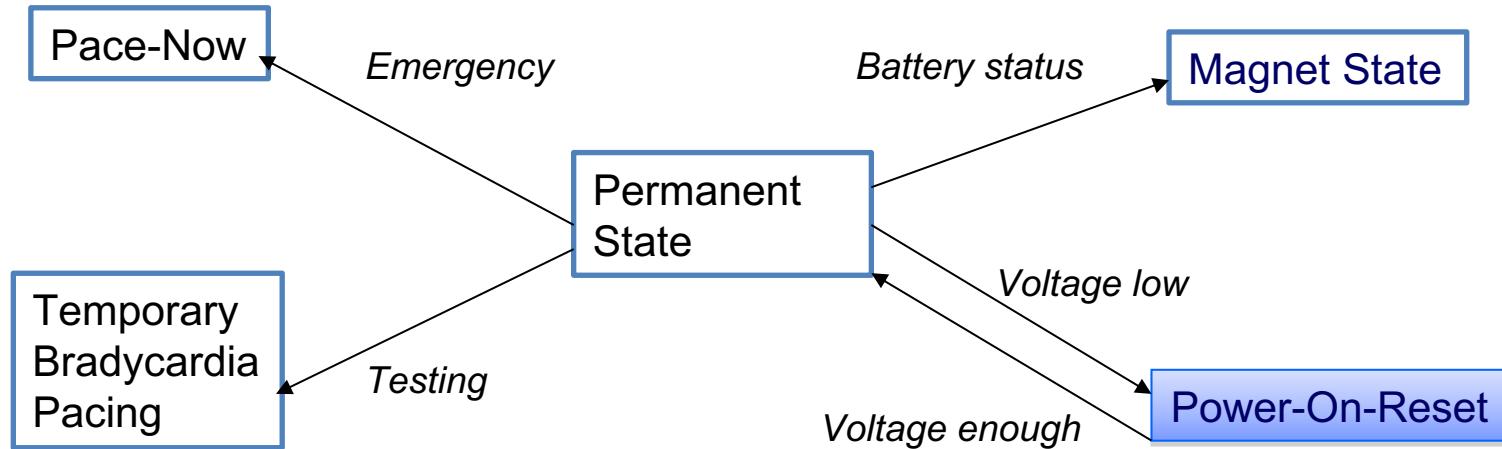
Pacemaker Operating States

Magnet State: Used during magnet testing to determine battery status of the device



Pacemaker Operating States

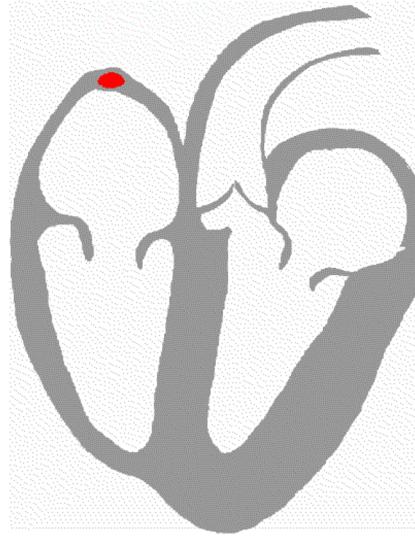
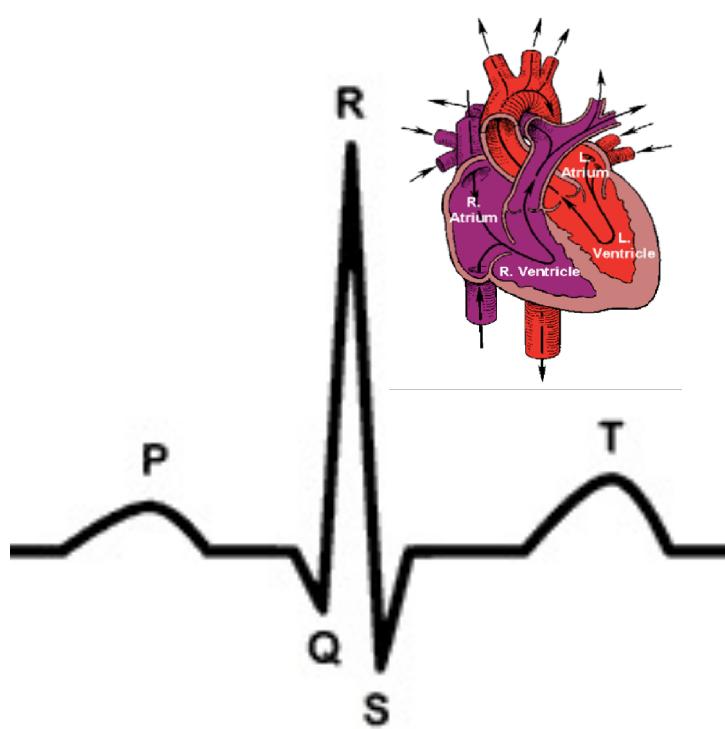
Power-On-Reset: Functions are disabled when battery voltage drops below a certain level, and state is resumed when the voltage is restored.



Contents

- I. Introduction to Pacemaker Challenge
- II. Background Information
- III. Pacemaker System
- IV. Pacemaker Timing Cycles
- V. Project Overview

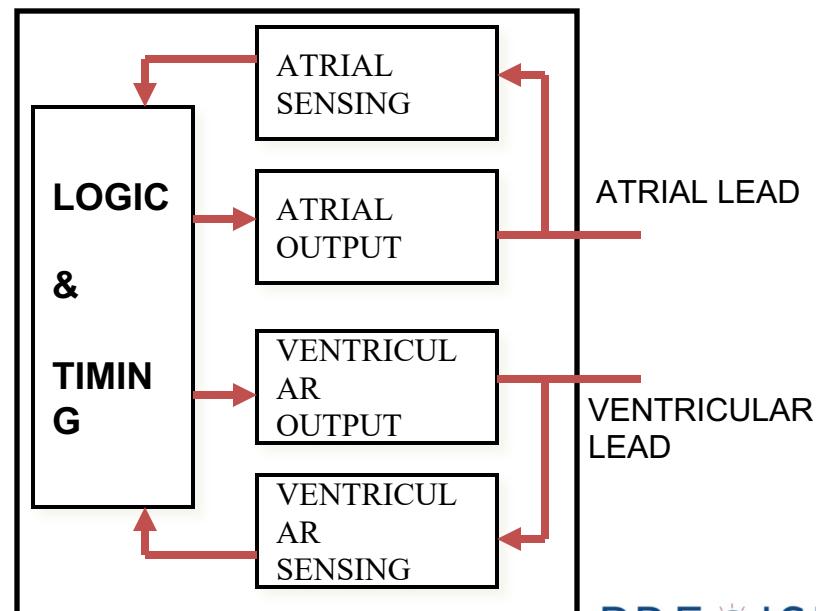
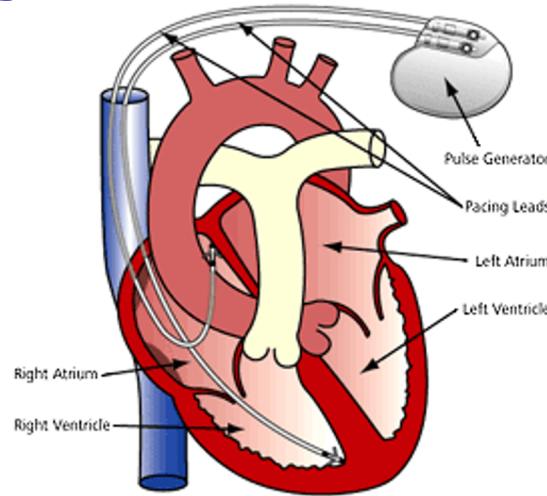
Reading ECG



- **P wave:** normal atrial depolarization
- **QRS complex**
 - Depolarization of the right and left ventricles
 - A recording of a single heartbeat on the ECG
- **T wave:** the repolarization (or recovery) of the ventricles.

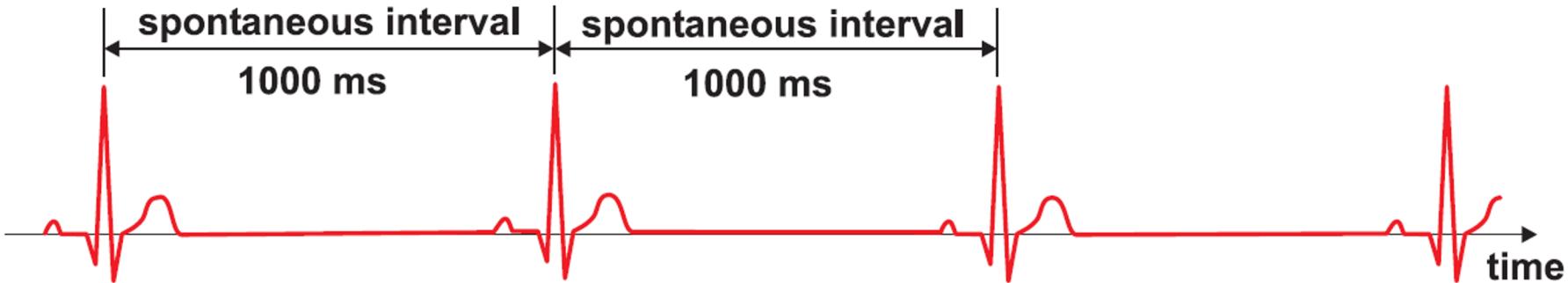
Timing Cycles

- Pacemaker senses and paces in both atrium and ventricle
- Synchronization between atrial and ventricular sensing and pacing
- Interference from atrial stimuli to ventricular channel and vice-versa.

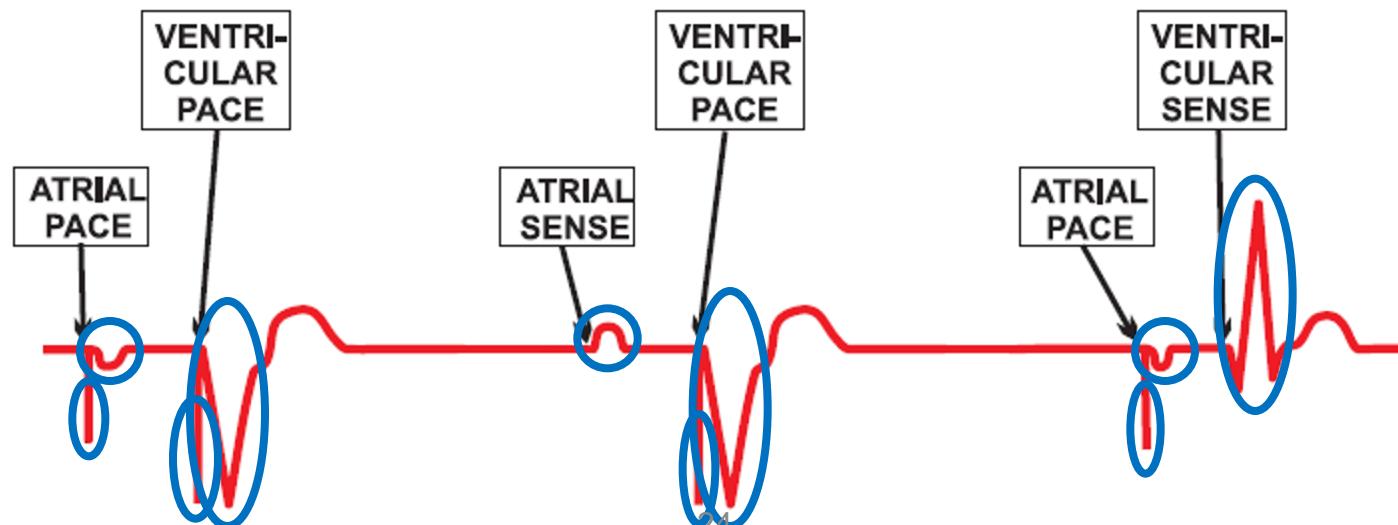


Reading ECG

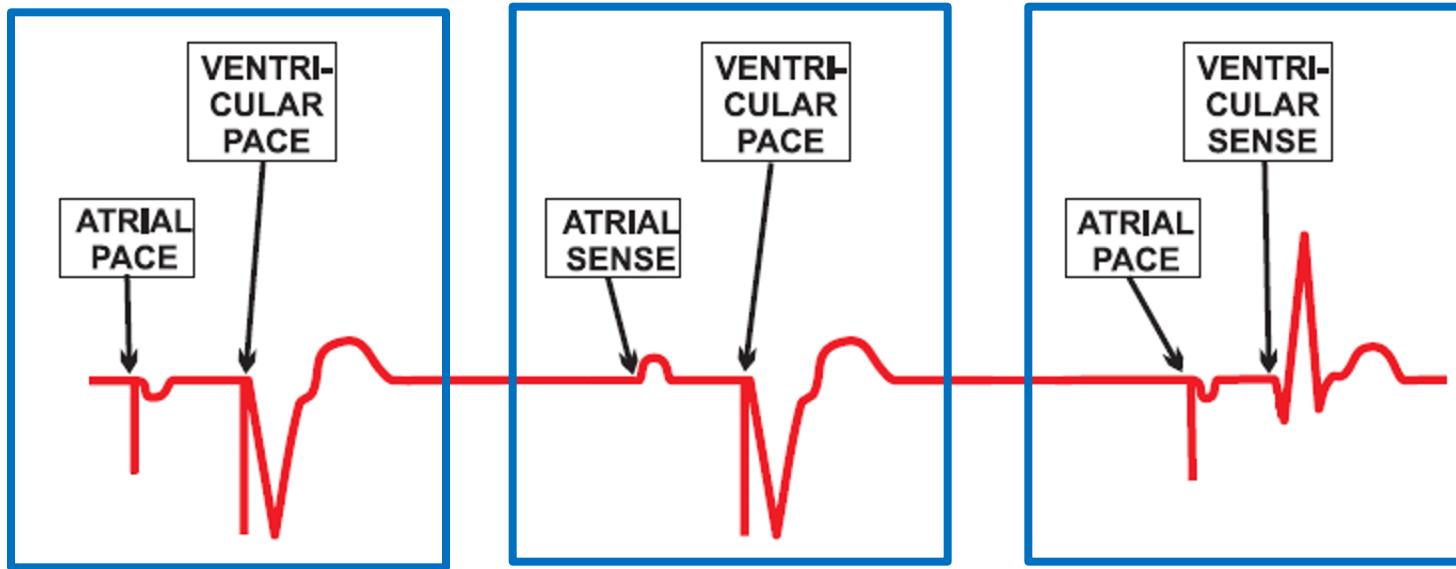
Normal Heart



Heart with problems



Reading ECG



- **No Sensing:** pacemaker delivers both atrial and ventricle signals
- **Atrial Sensing**
 - Atrial sensing inhibits scheduled atrial pacing
 - Pacemaker delivers ventricle pacing
- **Ventricle Sensing:** ventricle sensing inhibits scheduled ventricle pacing

Pacemaker Operating Modes

In the following slides and the project, we are concerned with the VVI pacemaker. Where VVI stands for **V**entricle being paced if the **V**entricle activity sensed is **I**nhibited.

I	II	III	IV
Chamber(s) paced	Chamber(s) sensed	Response to sensing	Rate adaptive
O = none	O = none	O = none	O = none
A = atrium	A = atrium	I = Inhibited	R = rate adaptive
V = ventricle	V = ventricle	T = Triggered	
D = dual	D = dual	D = dual	

VVI mode: if natural R wave is not sensed, pacemaker has to pace
Extra Credit Examples: VDD, DDD, VDDR, VVIR, VDDR, DDDR

Pacemaker Operating Modes

I	II	III	IV
Chamber(s) paced	Chamber(s) sensed	Response to sensing	Rate adaptive
O = none	O = none	O = none	O = none
A = atrium	A = atrium	I = Inhibited	R = rate adaptive
V = ventricle	V = ventricle	T = Triggered	
D = dual	D = dual	D = dual	

23 programmable pacing modes

- VOO: ventricle paced, no sensing (and no response to sensing)
- VVI: ventricle paced and sensed. Reaction to sensing a QRS is to inhibit the pace.
- DDD: both chambers paced and sensed. Sensed P and QRS can inhibit a pace, sensed P can trigger a ventricular pace.
- AAT: atrium paced, atrium sensed, pace triggered when atrium is sensed

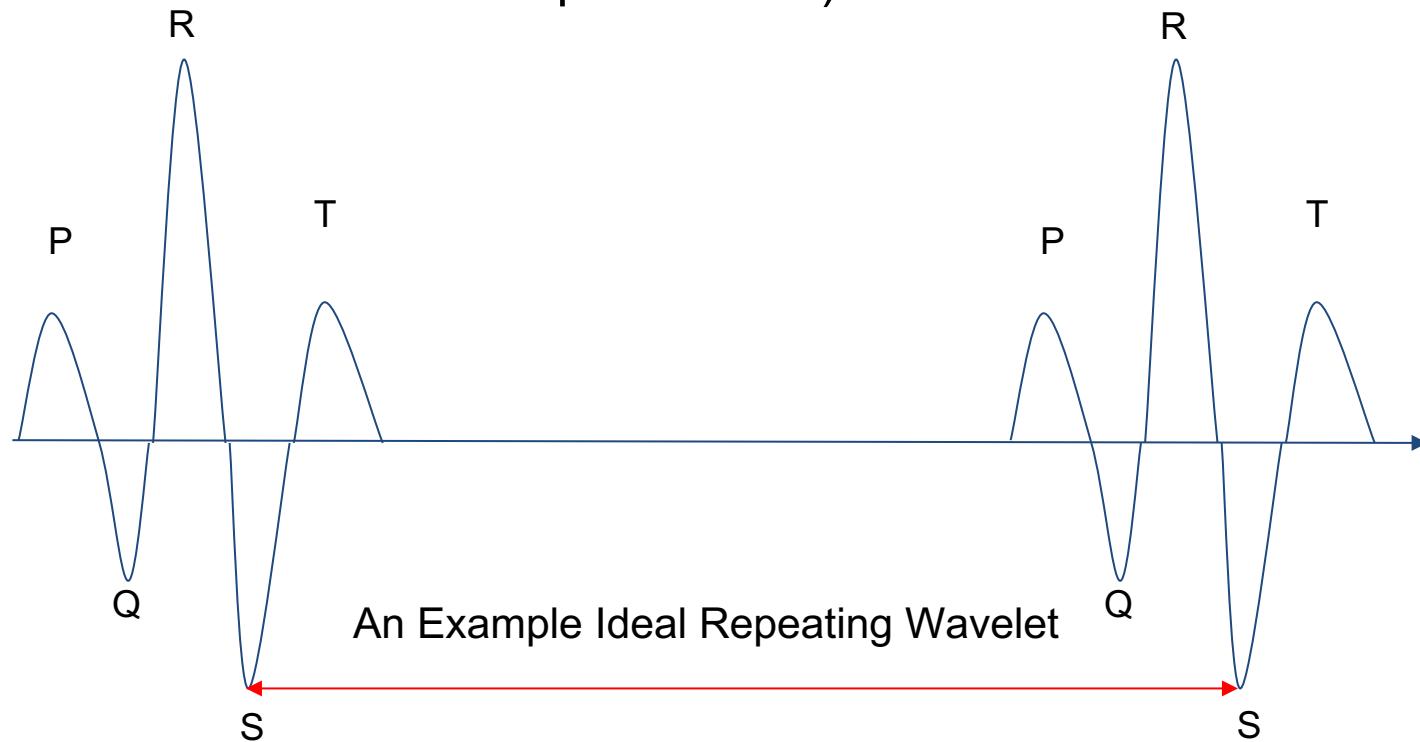
Pacemaker Operating Modes

- ❑ There are 10 non-rate-adaptive *modes*, each associated with a 3-letter acronym:
 - The first refers to the chamber(s) paced by the device: **V** (ventricle), **A** (atrium), **D** (dual), or **O** (neither)
 - The second refers to the chamber(s) in which the device senses, again **V**, **A**, **D**, or **O**.
 - The third refers to the pacemaker's response to sensing: **T** (triggers pacing), **I** (inhibits pacing), **D** (tracked pacing), or **O** (neither).
 - T: During triggered pacing, a sense in a chamber shall trigger an immediate pace in that chamber.
 - I: During inhibited pacing, a sense in a chamber shall inhibit a pending pace in that chamber.
 - D: During tracked pacing, an atrial sense shall cause a tracked ventricular pace after a programmed AV delay, unless a ventricular sense was detected beforehand.

The VVI mode Pacemaker

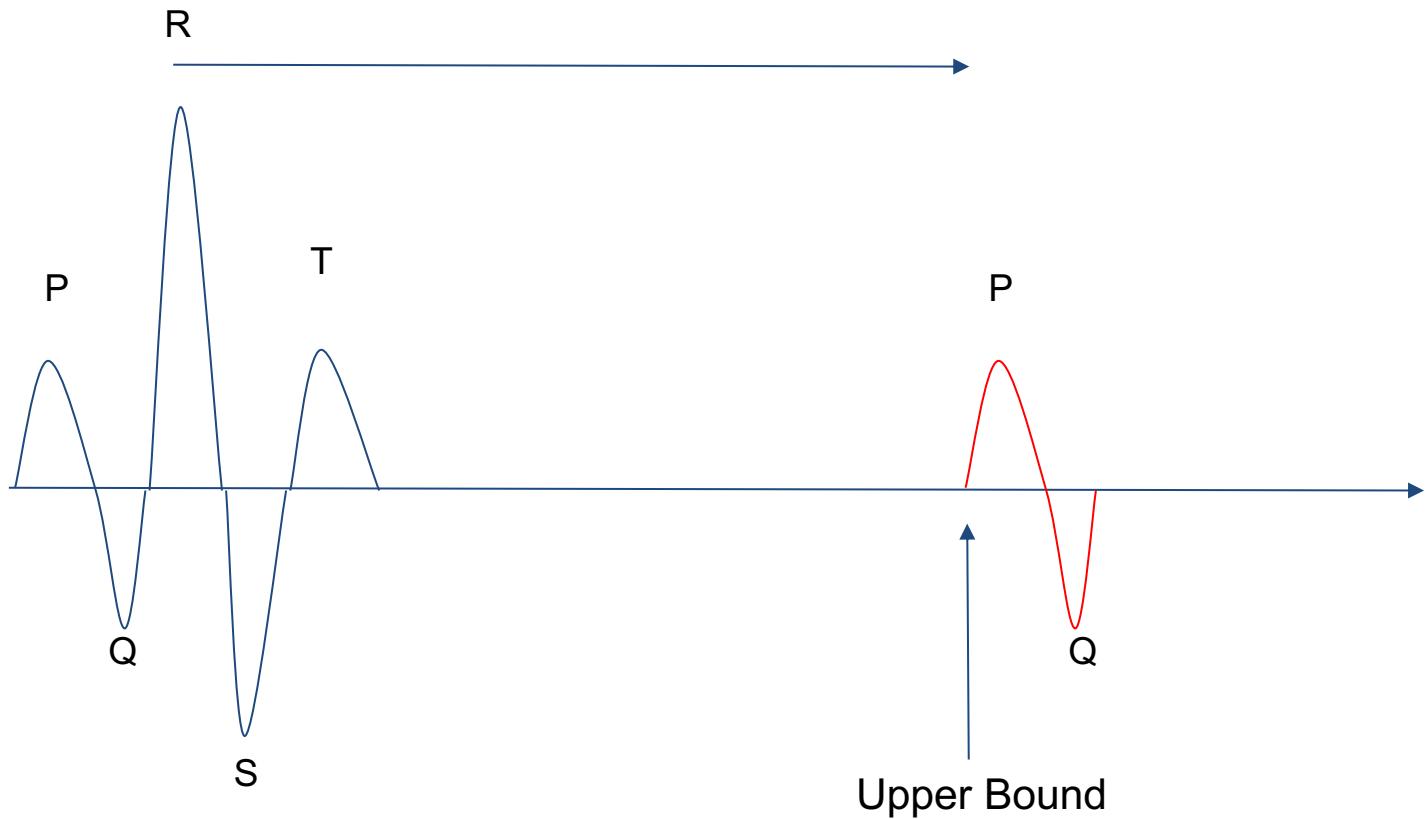
The pacemaker should be inactive for a healthy ECG signal.

(The PQRST waves in the following slides and in the project have been simplified for ease of understanding the workings of a VVI mode pacemaker)



The VVI mode Pacemaker

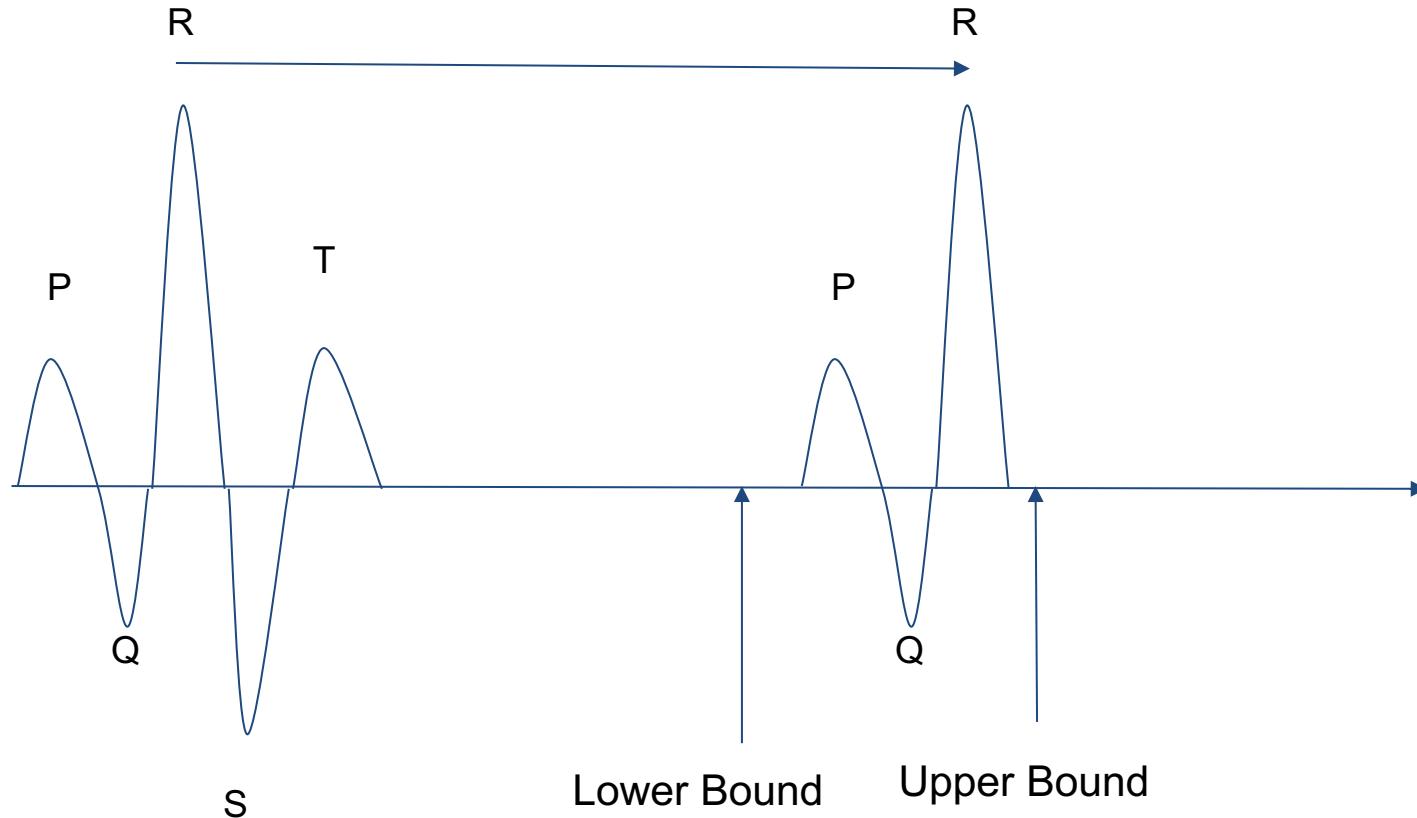
The pacemaker should pace if the R-R interval exceeds an upper bound



Pacemaker paces here to force-restart RandomHeart.

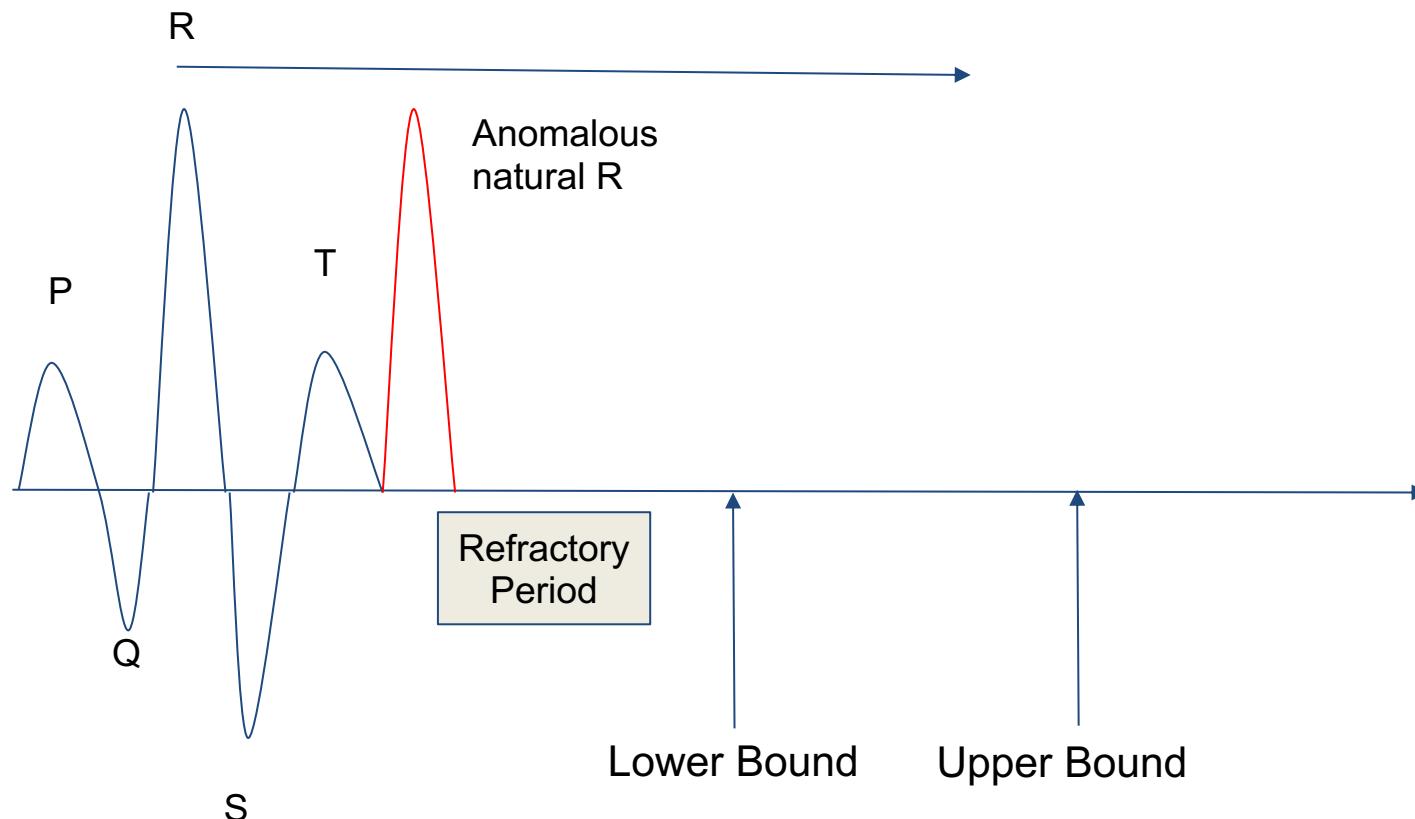
The VVI mode Pacemaker

The pacemaker should not pace if a natural R wave occurs at an R-R interval greater than the lower bound



The VVI mode Pacemaker

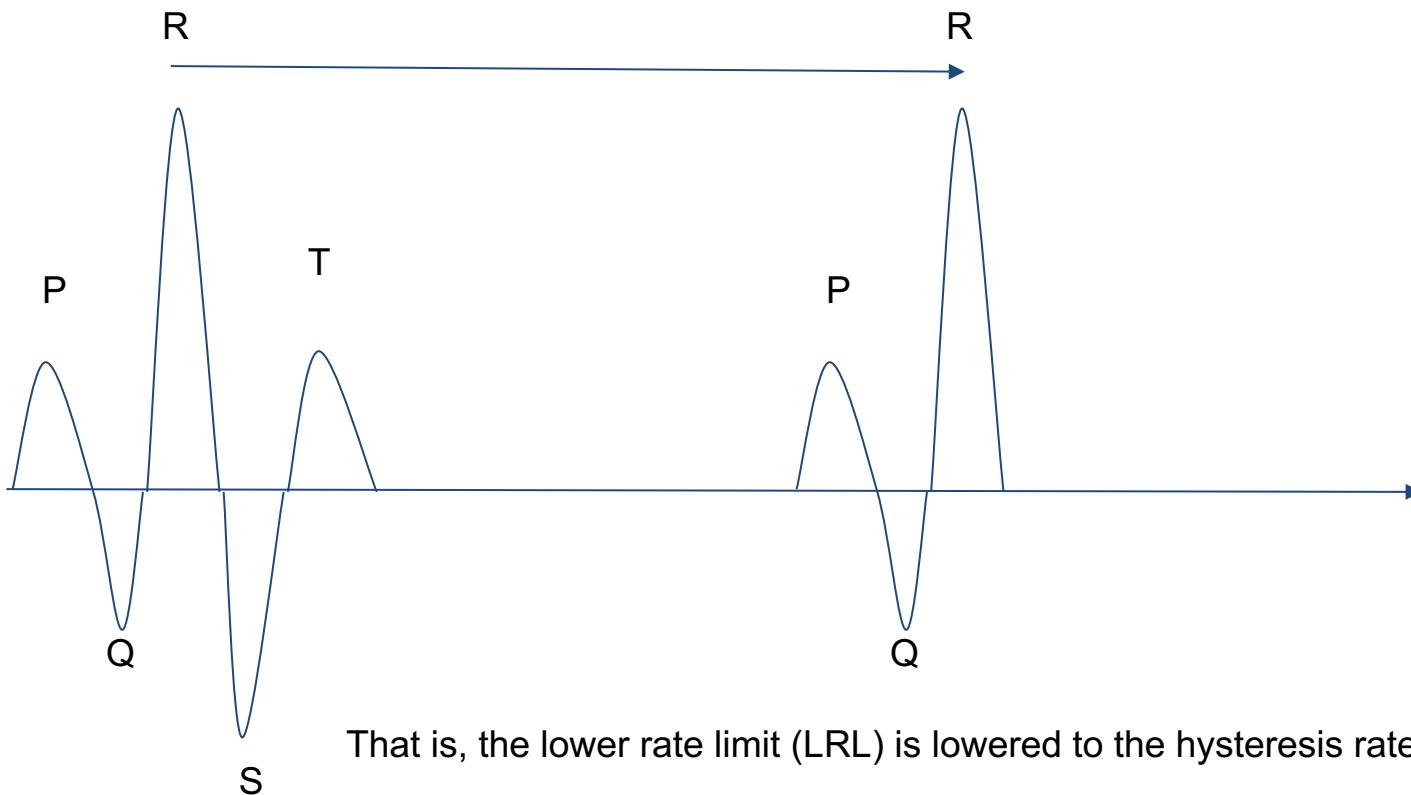
If an anomalous R wave occurs within the VRP (ventricular Refractory Period) after a PQRST, said R wave should be ignored. The R-R interval needs to still be calculated from the earlier R wave.



Hysteresis

Hysteresis tries to encourage the heart to beat naturally (without pace).

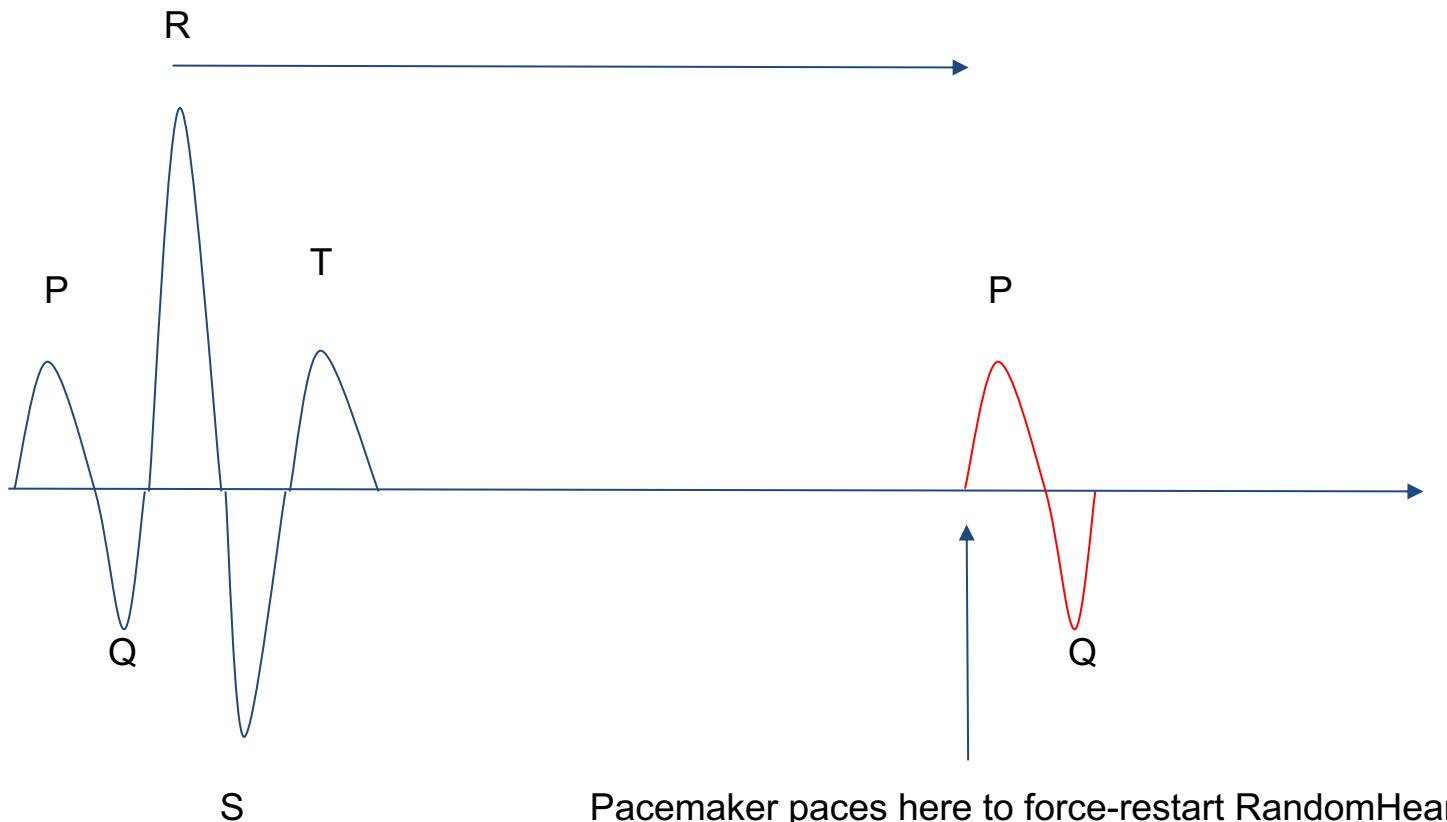
In the following situation, when the heart beats naturally, hysteresis is enabled and the pacemaker allows for a longer interval after sensing an R wave to pace the heart. Otherwise, hysteresis is disabled.



That is, the lower rate limit (LRL) is lowered to the hysteresis rate limit (HRL).

Hysteresis

But if the upper limit is reached, the pacemaker paces.



We can have more cases

The Heart can combine the previously discussed cases. Pacemaker should consider all timing constraints carefully.

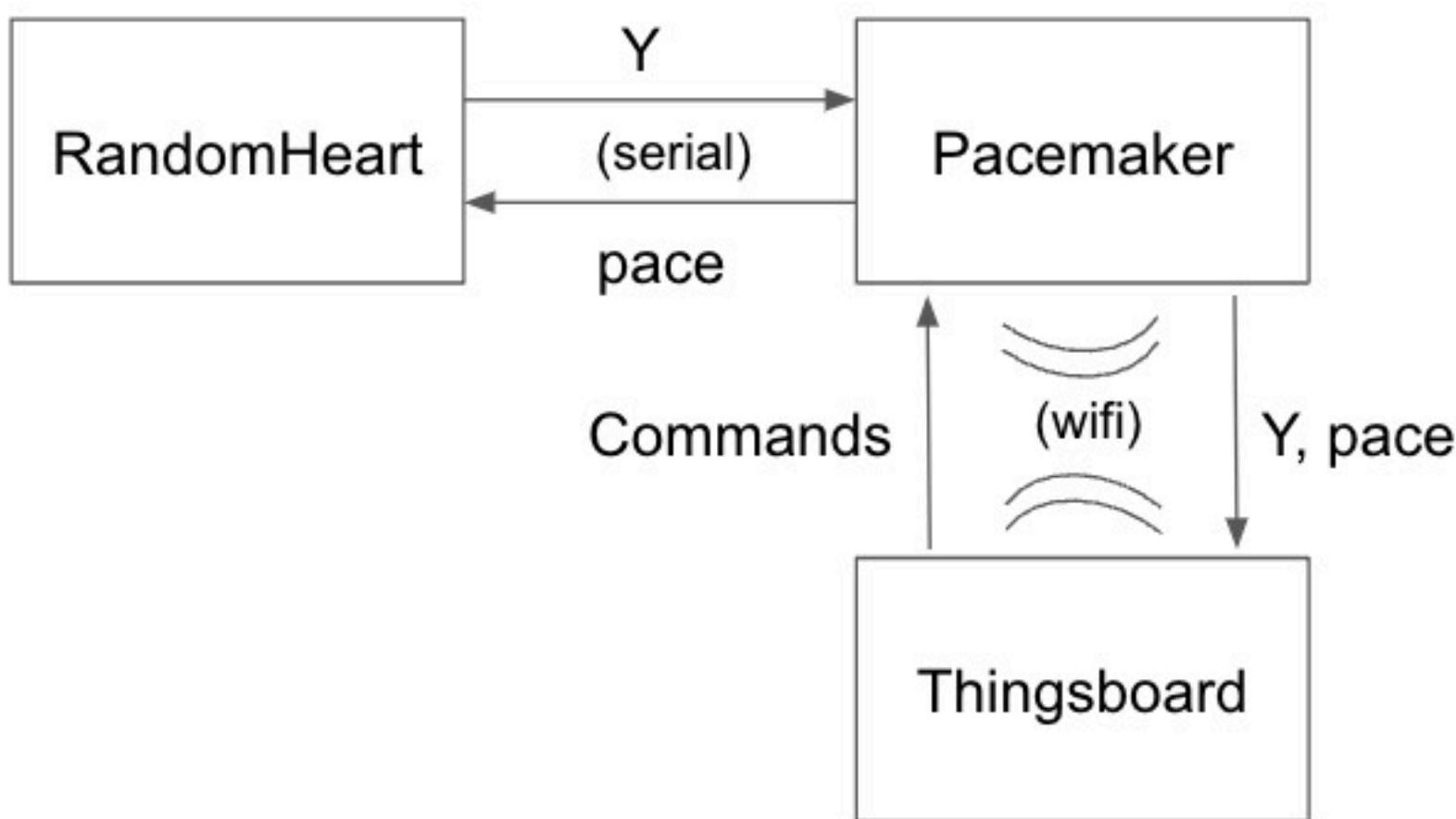
Contents

- I. Introduction to Pacemaker Challenge
- II. Background Information
- III. Pacemaker System
- IV. Pacemaker Timing Cycles
- V. Project Overview

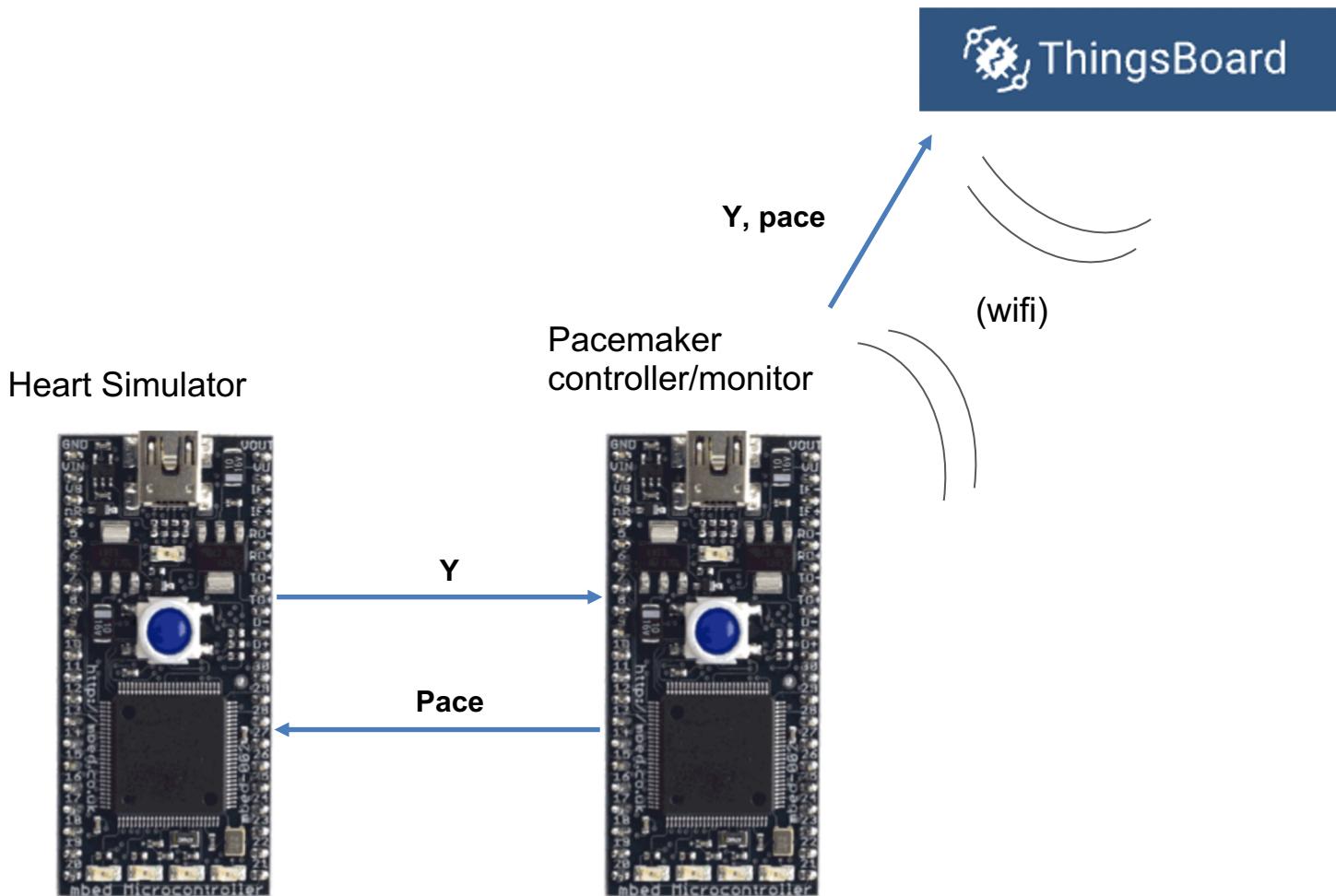
Milestones

1. Implementation of the Smart Pacemaker
 - a. Heart and PacemakerMBEDs (20%)
 - b. Thingsboard and Remote Control (10%)
2. Verification and Security of the Smart Pacemaker
 - a. Verification (20%)
 - b. Security (20%)
3. Validation, Demo, and Final Report
 - a. Validation with Assurance Cases (20%)
 - b. Teamwork and Member Contribution (5%)
 - c. Demo (5%)

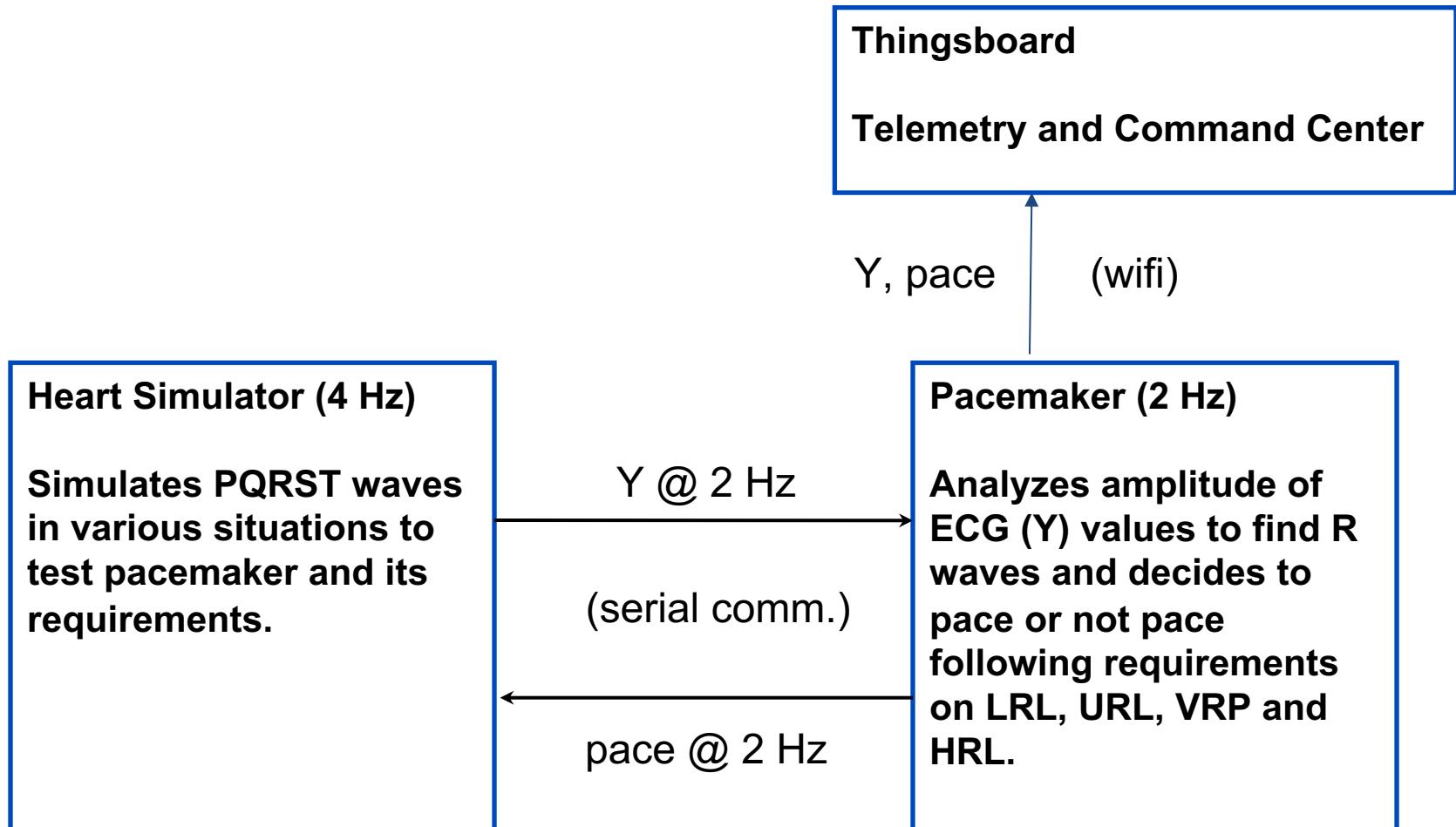
Milestone 1: Implementation of the Smart Pacemaker



Implementation Scheme



Overview (is there chat?)



Milestone 2: Verification, and Security of the Smart Pacemaker

1. Verify Pacemaker Properties to make sure the Design is Conceptually Correct
 1. UPAAL- an integrated tool environment for modeling, validation and verification of real-time systems
2. Secure the Thingsboard Communication Layer
 - a. Existing MQTT Security
 - b. Cryptographic Protocol(s)

Milestone 3: Validation, Demo, and Final Report

1. Develop assurance case to argue the correctness of your system
2. Final Report
 1. Implementation Architecture
 2. Design Models and Correctness Properties
 3. Assurance Cases
3. In-Class Demo

Smart Pacemaker Project

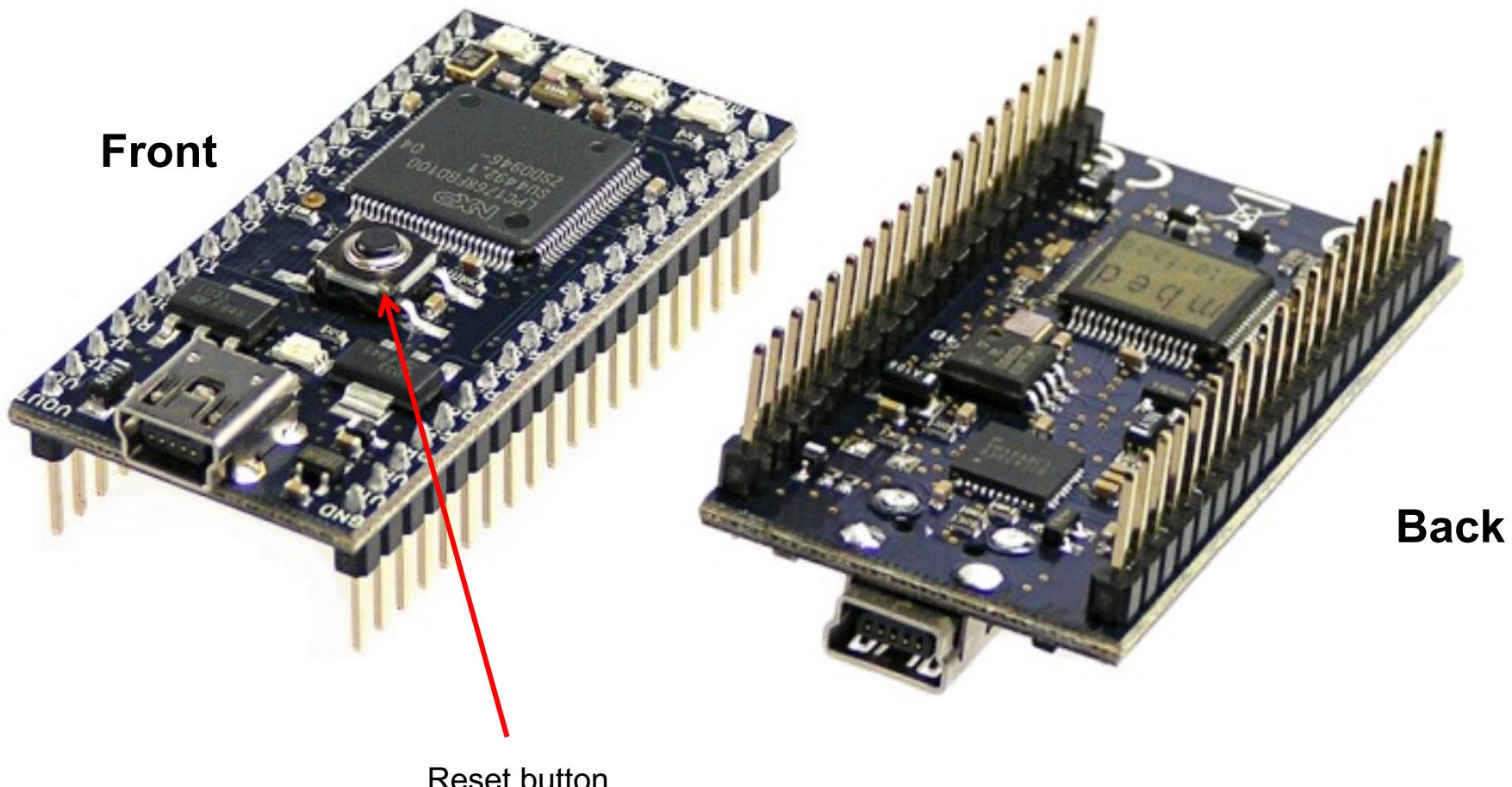
- Team project
- Based on the pacemaker challenge problem
- Project phases
 - Milestone 1: Implementation
 - Milestone 2: Verification and Security
 - Milestone 3: Demo and Final Report

Acknowledgements

- Slides Created By:
 - Kausthubh Sridhar

Introduction to MBED

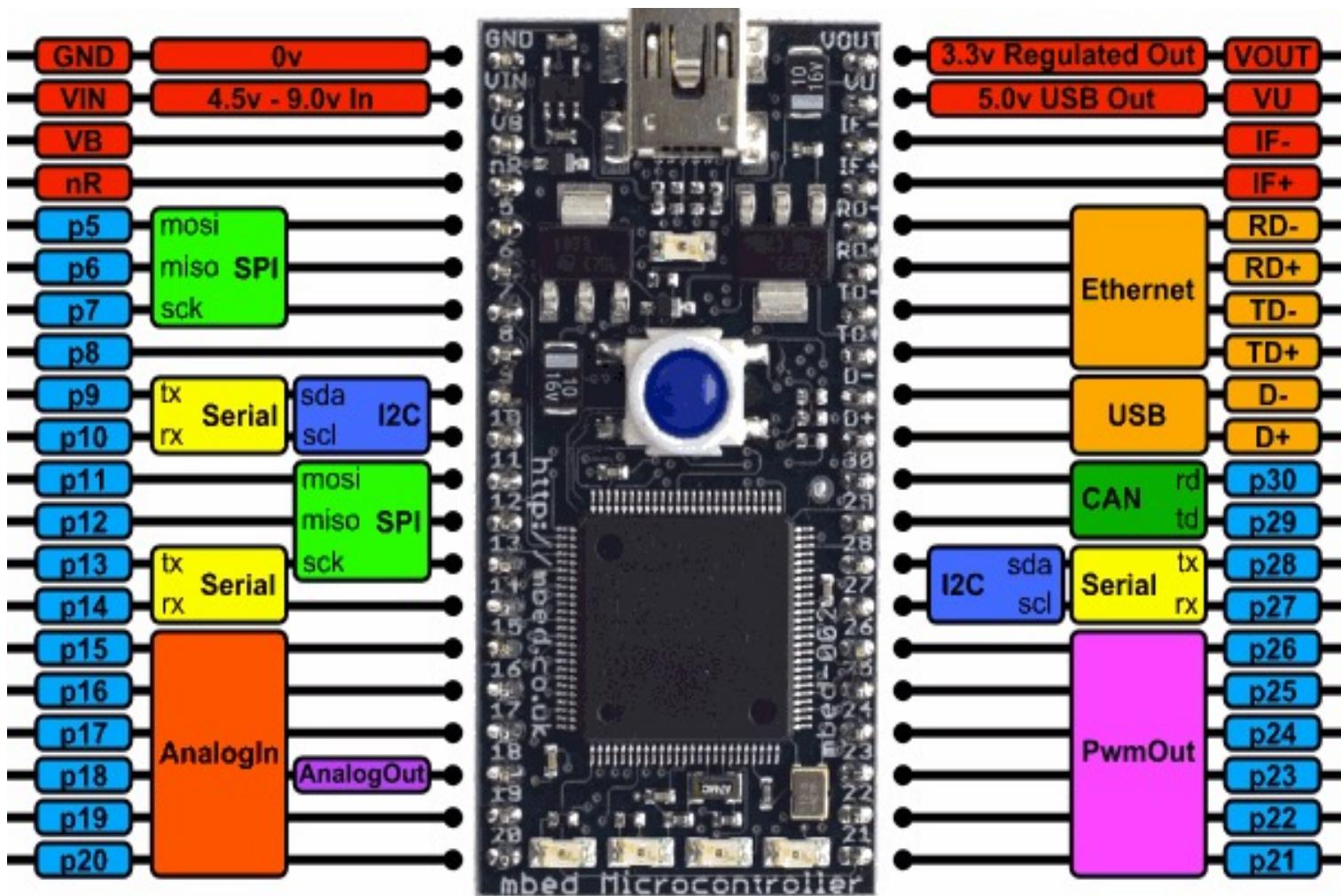
mbed LPC1768



mbed 1768 features

- LPC1768 MCU
 - High performance ARM® Cortex™-M3 Core
 - 96MHz, 32KB RAM, 512KB FLASH
 - Ethernet, USB Host/Device, 2xSPI, 2xI2C, 3xUART, CAN, 6xPWM, 6xADC, GPIO
- Other features
 - 40-pin, 54x26mm
 - 5V USB or 4.5-9V supply
 - Built-in USB drag 'n' drop FLASH programmer

mbed 1768 features



Programming on mbed

- Website: <http://mbed.org/>
 - Platform, components, handbook, cookbook, code, questions, Forum, Compile
- Mbed 1768: <http://mbed.org/platforms/mbed-LPC1768/>
- Mbed OS 5: <https://os.mbed.com/docs/mbed-os/v5.13>
- TextLCD: <http://mbed.org/cookbook/Text-LCD-Enhanced>
- Tutorial: <https://os.mbed.com/docs/mbed-os/v5.13/tutorials/serial-communication.html>
- I/O tutorial, general tutorial
- Books: <http://mbed.org/cookbook/Books>

Programming on MBED

- Register and login
- Set up the hardware
- Create a new project in mbed compiler
- Write a hello world program
- Compile
- Downloading to mbed



Register and Login

Register and login

Set up the hardware

Create a new project in mbed compiler

Write a hello world program

Compile

Downloading to mbed

The screenshot shows the official website for Arm Mbed. At the top, there's a navigation bar with icons for search, user profile, and settings. Below the header, the "Compiler" button is highlighted with a red box. The main content area features the "Mbed" logo and the tagline "Rapid IoT device development". A descriptive paragraph explains that Mbed provides a free open-source IoT operating system with connectivity, security, storage, device management, and machine learning. To the right, there's a graphic illustrating the Mbed ecosystem, showing a laptop connected to a cloud icon, which is then connected to a lock icon and a circular interface. Below the laptop is a speaker icon, and further down is a building icon.

Register and Login

Register and login

Set up the hardware

Create a new project in mbed compiler

Write a hello world program

Compile

Downloading to mbed

The screenshot shows the mbed online compiler interface. The top navigation bar includes 'New', 'Import', 'Save', 'Save All', 'Compile', 'Commit', and 'Revisions' buttons. The left sidebar is titled 'Program Workspace' and shows a tree view of 'My Programs'. Under 'LCD', there is a 'TextLCD' folder containing a 'main.cpp' file, which is currently selected. Other programs listed are 'mbed', 'LCDTimer', and 'timer'. The main right pane displays the contents of 'main.cpp':

```
1 #include "mbed.h"
2 #include "TextLCD.h"
3
4 TextLCD lcd(p15, p16, p17, p18, p19, p20,
5
6 int main() {
7     lcd.printf("Hello World!\n");
8     lcd.printf("Hello World!\n");
9 }
```

Setup the Hardware

Register and login

Set up the hardware

Create a new project in mbed compiler

Write a hello world program

Compile

Downloading to mbed



Setup the Hardware - LCD

Register and login

Set up the hardware

Create a new project in mbed compiler

Write a hello world program

Compile

Downloading to mbed

LCD pins	mbed pins
p1 GND	p1 0V
p2 VCC	p39 5V VU
p3 VO via 2.2kΩ resistor	p1 0V
p4 RS	p15
p5 RW	p1 0V
p6 E	p16
p11 D4	p17
p12 D5	p18
p13 D6	p19
p14 D7	p20

Setup the Hardware

Register and login

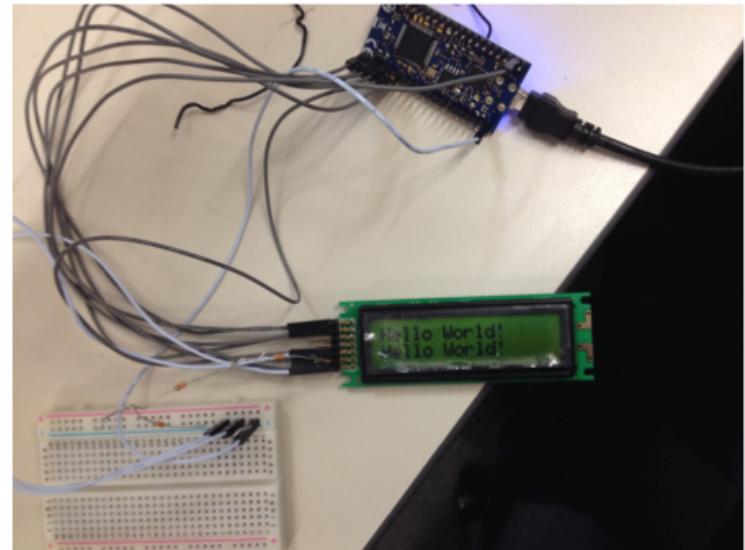
Set up the hardware

Create a new project in mbed compiler

Write a hello world program

Compile

Downloading to mbed



Create a New Project

Register and login

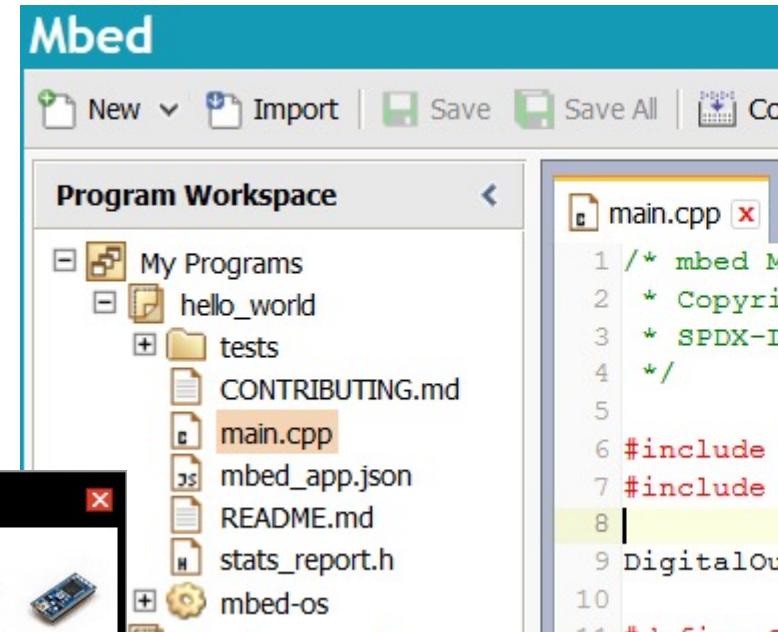
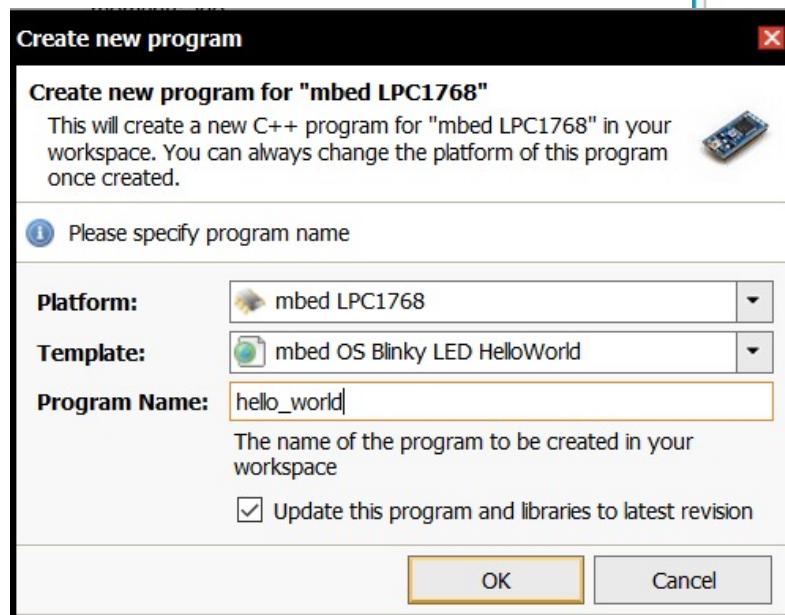
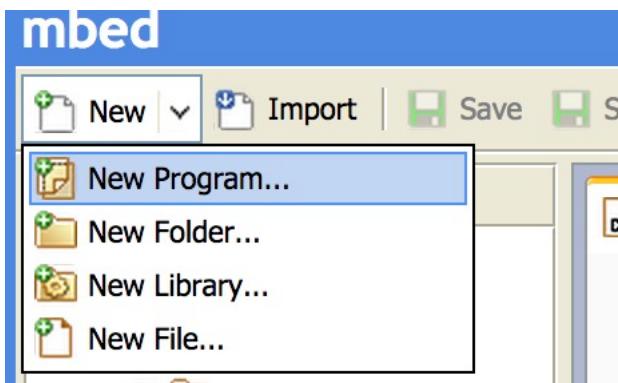
Set up the hardware

Create a new project in mbed compiler

Write a hello world program

Compile

Downloading to mbed



Import Required Libraries

Register and login

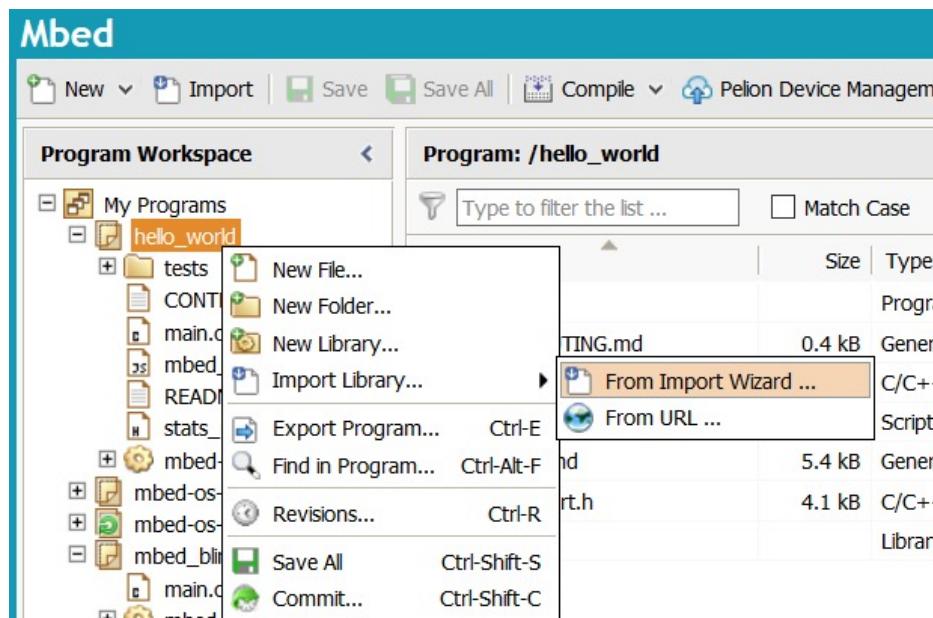
Set up the hardware

Create a new project in mbed compiler

Write a hello world program

Compile

Downloading to mbed



Search "TextLCD"

Programs	Libraries	Bookmarked	Upload	TextLCD	Search												
Listing published libraries on os.mbed.com matching "TextLCD"																	
<table><thead><tr><th>Name</th><th>Tags</th><th>Author</th><th>Imports</th><th>Modified</th><th>Description</th></tr></thead><tbody><tr><td>★ TextLCD</td><td>HD44780 TextLCD</td><td>Simon Ford</td><td>68585</td><td>02 Jan 2014</td><td>TextLCD library for controlling various LCD panels based on the HD44780 4-bit interface</td></tr></tbody></table>						Name	Tags	Author	Imports	Modified	Description	★ TextLCD	HD44780 TextLCD	Simon Ford	68585	02 Jan 2014	TextLCD library for controlling various LCD panels based on the HD44780 4-bit interface
Name	Tags	Author	Imports	Modified	Description												
★ TextLCD	HD44780 TextLCD	Simon Ford	68585	02 Jan 2014	TextLCD library for controlling various LCD panels based on the HD44780 4-bit interface												

Hello World!

Register and login

Set up the hardware

Create a new project in mbed compiler

Write a hello world program

Compile

Downloading to mbed

```
1 #include "mbed.h"
2 #include "TextLCD.h"
3
4 TextLCD lcd(p15, p16, p17, p18, p19, p20, TextLCD::LCD16x2); // rs, e, d4-d7
5
6 int main() {
7     lcd.printf("Hello World!\n");
8     lcd.printf("Hello World!\n");
9 }
```

Compile

Register and login

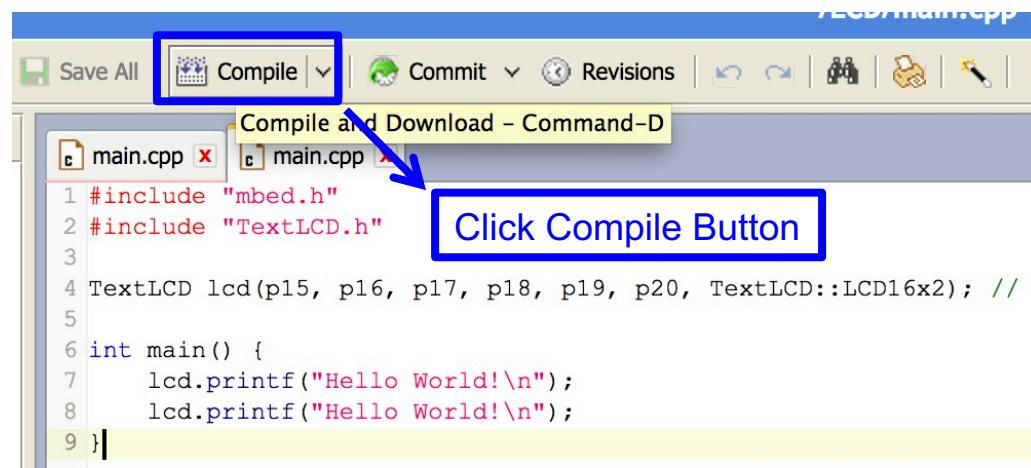
Set up the hardware

Create a new project in mbed compiler

Write a hello world program

Compile

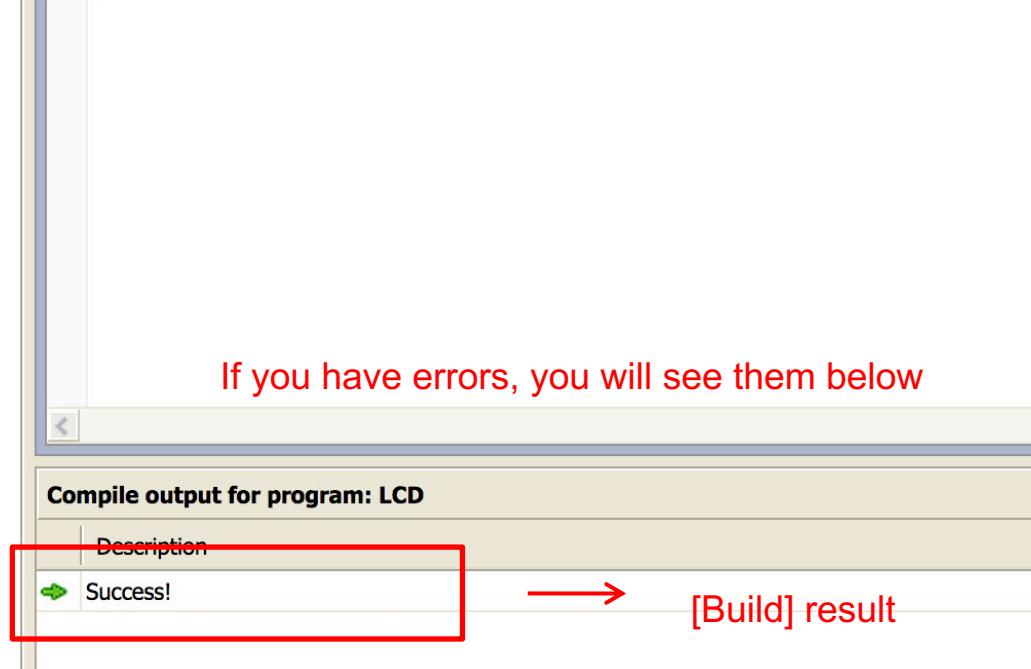
Downloading to mbed



Click Compile Button

```
1 #include "mbed.h"
2 #include "TextLCD.h"
3
4 TextLCD lcd(p15, p16, p17, p18, p19, p20, TextLCD::LCD16x2); // i
5
6 int main() {
7     lcd.printf("Hello World!\n");
8     lcd.printf("Hello World!\n");
9 }
```

If you have errors, you will see them below



Description
Success!

[Build] result

Download to MBED

Register and login

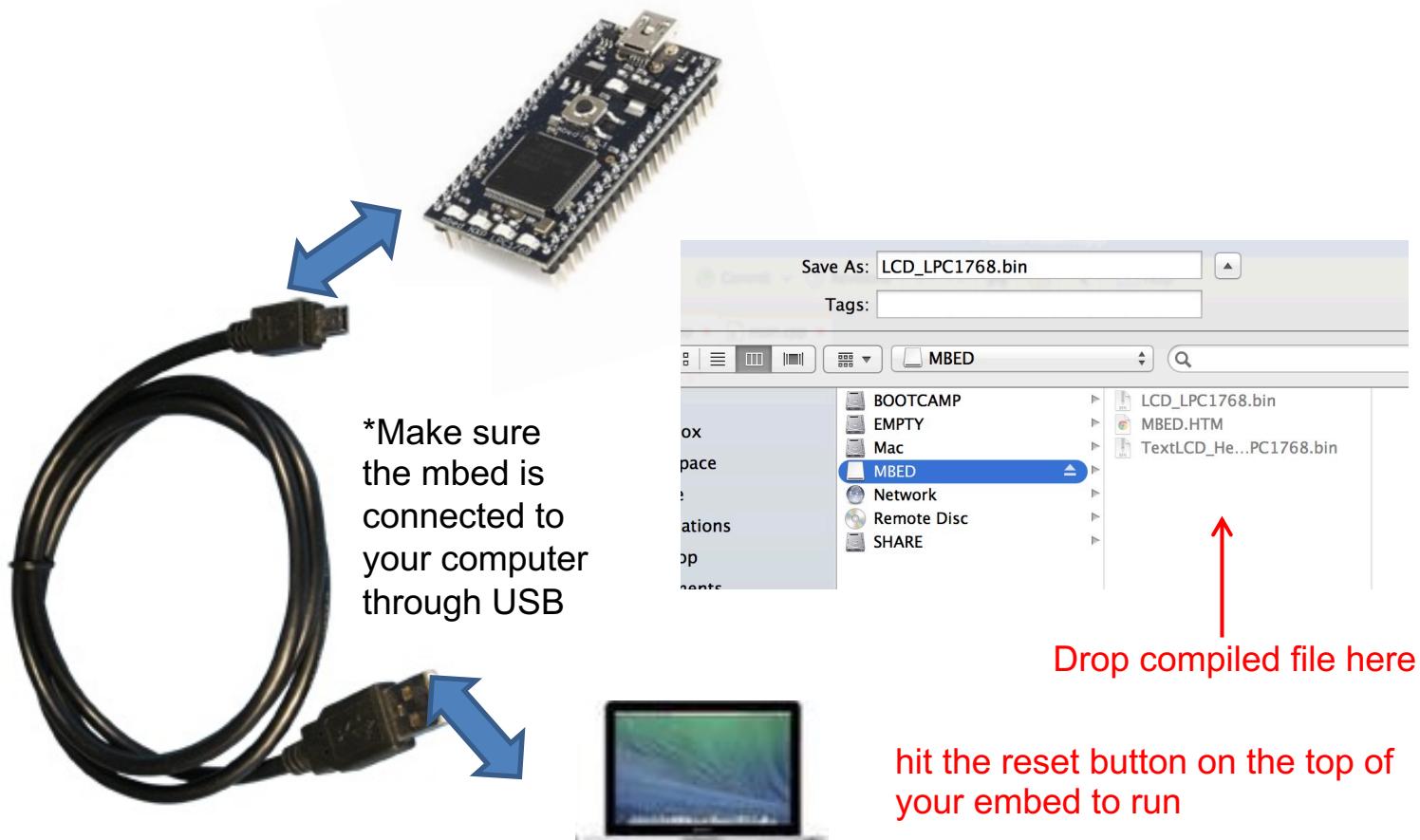
Set up the hardware

Create a new project in mbed compiler

Write a hello world program

Compile

Downloading to mbed



*If you don't have an MBED.HTML file, <https://os.mbed.com/platforms/mbed-LPC1768/>

Viewing Output in Terminal

Register and login

Set up the hardware

Create a new project in mbed compiler

Write a hello world program

Compile

Downloading to mbed

- MBED CLI1 to view output in terminal
 - Install python 3 (pip install python3)
 - Install mbed-cli (pip install mbed-cli)
 - (Additional instructions in
<https://os.mbed.com/docs/mbed-os/v6.15/build-tools/install-and-set-up.html>)
 - To verify Mbed CLI installed correctly, run
mbed -help
 - To open a serial terminal (only work for a single board) for message printing, run
mbed sterm -b <BAUDRATE>
(default 9600)

Code Reuse

- Programming microcontrollers for a simple task is mostly about setting up the right registers in the right way; however, neither is straightforward.
- Don't need to start from scratch for everything.
- It's OK to learn how to initialize, set up timer/interrupts, and write LCD from code that is already available.
- Be sure to cite/ack the sources.

Acknowledgements

- Slides Created By:
 - Dagaen Golomb
 - Yahan Yang
 - Amanda Watson

Introduction to Thingsboard

ThingsBoard is at <https://tb-exp.precise.seas.upenn.edu/>

Once you have a team, please contact Amanda Watson (aawatson@seas.upenn.edu) to create an account with your team name and emails for you and your teammates. She will give you access to your group's Thingsboard.

Home Page

The screenshot shows the ThingsBoard home page with a dark blue header and sidebar. The header includes the ThingsBoard logo and a 'Home' button. The sidebar contains a list of navigation items: Home, Rule chains, Customers, Assets, Devices, Device profiles, OTA updates, Entity Views, Edge instances, Edge management, Widgets Library, Dashboards, Audit Logs, and API Usage. The main content area is divided into six cards:

- Rules management**: Icon of two overlapping arrows, labeled "Rule chains".
- Customer management**: Icon of two people, labeled "Customers".
- Asset management**: Icon of a grid, labeled "Assets".
- Device management**: Icon of a device with a gear, labeled "Devices".
- Device profiles**: Icon of a device with a 'D' inside, labeled "Device profiles".
- OTA updates**: Icon of a microchip, labeled "OTA updates".

Once you have a Thingsboard account, your homepage should look like this.

Device Creation

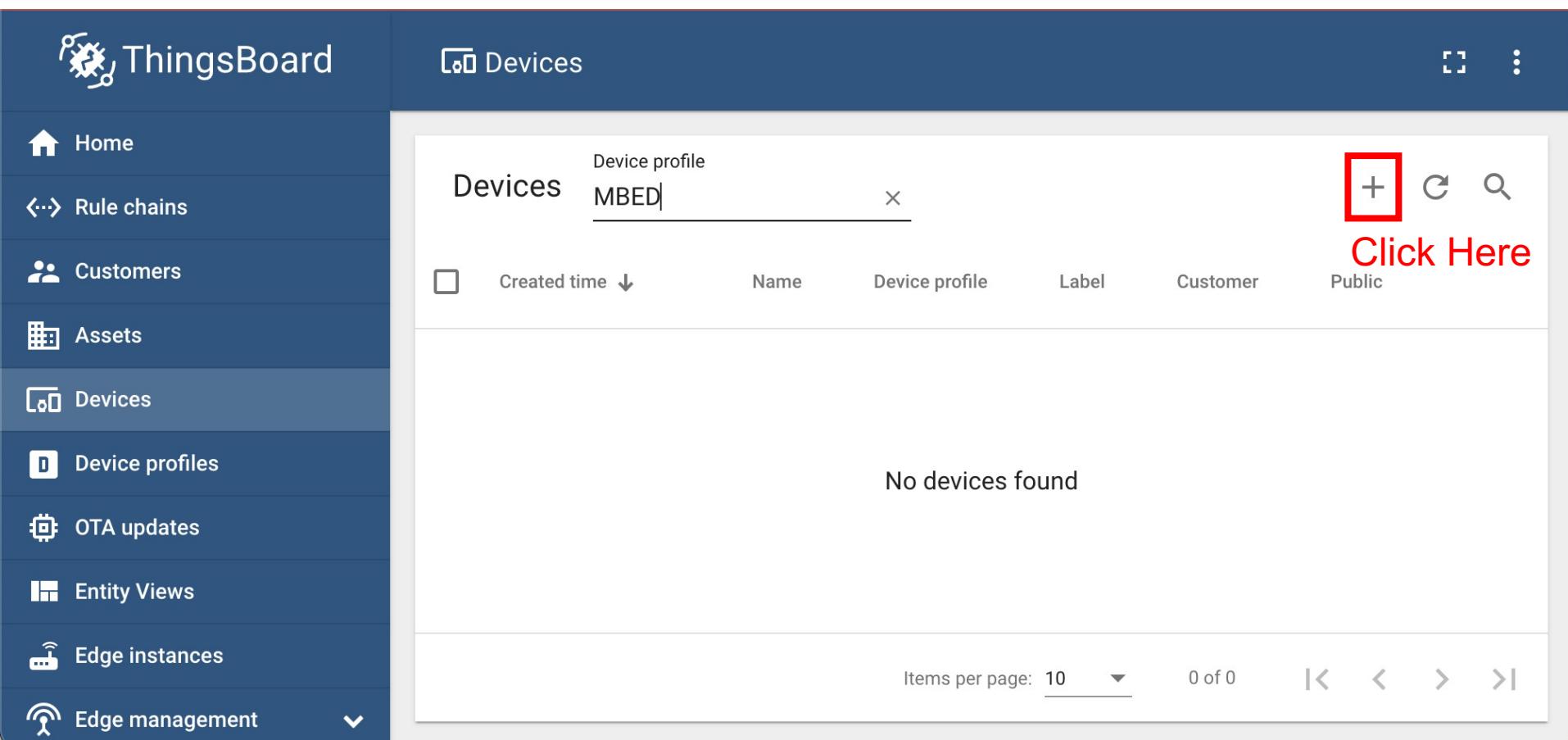
Device Creation

The screenshot shows the ThingsBoard IoT platform interface. On the left is a sidebar with various navigation options. The main area is titled "Home" and contains several management sections:

- Rules management**: Contains a "Rule chains" button.
- Customer management**: Contains a "Customers" button.
- Asset management**: Contains an "Assets" button.
- Device management**: Contains a "Devices" button, which is highlighted with a red border.
- Device profiles**: Contains a "Device profiles" button.
- OTA updates**: Contains an "OTA updates" button.

A large red text overlay "Click Here" is positioned at the bottom left of the highlighted "Devices" button.

Device Creation



The screenshot shows the ThingsBoard web interface. The left sidebar has a dark blue background with white icons and text. The 'Devices' icon is highlighted, indicating the current page. Other options include Home, Rule chains, Customers, Assets, Device profiles, OTA updates, Entity Views, Edge instances, and Edge management. The main content area has a light gray background. At the top, it says 'Devices' and 'Device profile' with a search bar containing 'MBED'. To the right are three icons: a red-bordered '+' for creating new devices, a circular arrow for refresh, and a magnifying glass for search. Below this is a red button labeled 'Click Here'. A table header row is shown with columns: Name, Device profile, Label, Customer, and Public. The table body below contains the message 'No devices found'. At the bottom, there are pagination controls for items per page (set to 10), a total count of 0 of 0, and navigation arrows.

ThingsBoard

Home

Rule chains

Customers

Assets

Devices

Device profiles

OTA updates

Entity Views

Edge instances

Edge management

Devices

Device profile

MBED

Created time ↓

Name

Device profile

Label

Customer

Public

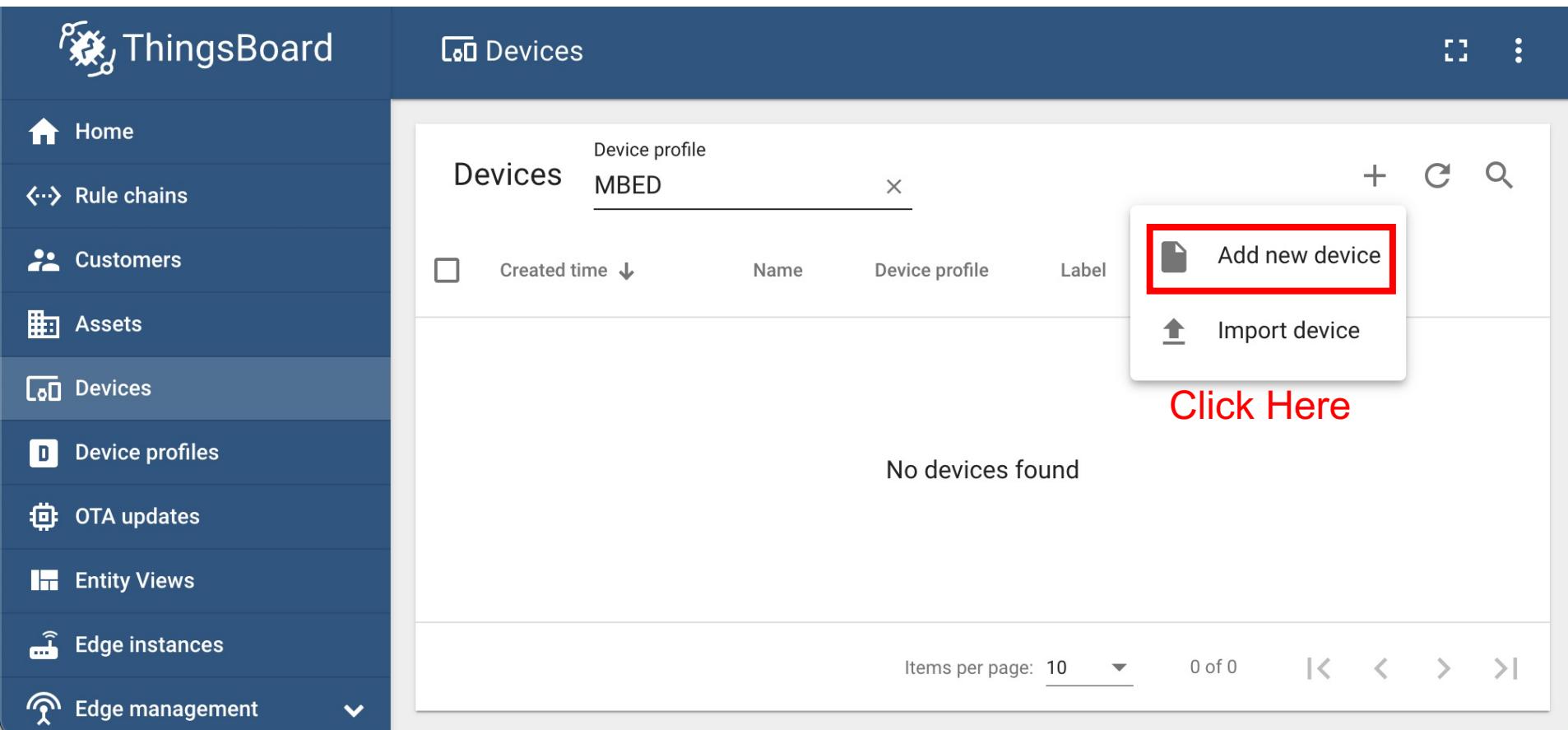
No devices found

Items per page: 10

0 of 0

Click Here

Device Creation



The screenshot shows the ThingsBoard web interface. The left sidebar has a dark blue background with white icons and text. The main area has a light gray background with a dark blue header bar.

ThingsBoard is displayed at the top left. The header bar includes a search icon and a three-dot menu icon.

The main title is **Devices**. Below it, a sub-header says **Device profile** followed by **MBED**. There is a close button (**x**) next to the profile name.

The table header contains columns: **Devices**, **Name**, **Device profile**, and **Label**. A checkbox and a sorting arrow are also present in the header.

A red box highlights the **Add new device** button, which has a file icon and the text "Add new device". Below it is another button with an upward arrow icon and the text "Import device".

A red button labeled "Click Here" is positioned below the "Import device" button.

The message "No devices found" is centered in the table area.

At the bottom, there are pagination controls: "Items per page: 10" with a dropdown arrow, "0 of 0", and navigation arrows (left, right, first, last).

Device Creation

Add new device

1 Device details 2 Credentials Optional 3 Customer Optional

Name *
Pacemaker **Add a Unique Name**

Label **Add a Label Here for Legends Graphs etc**

Select existing device profile **default**

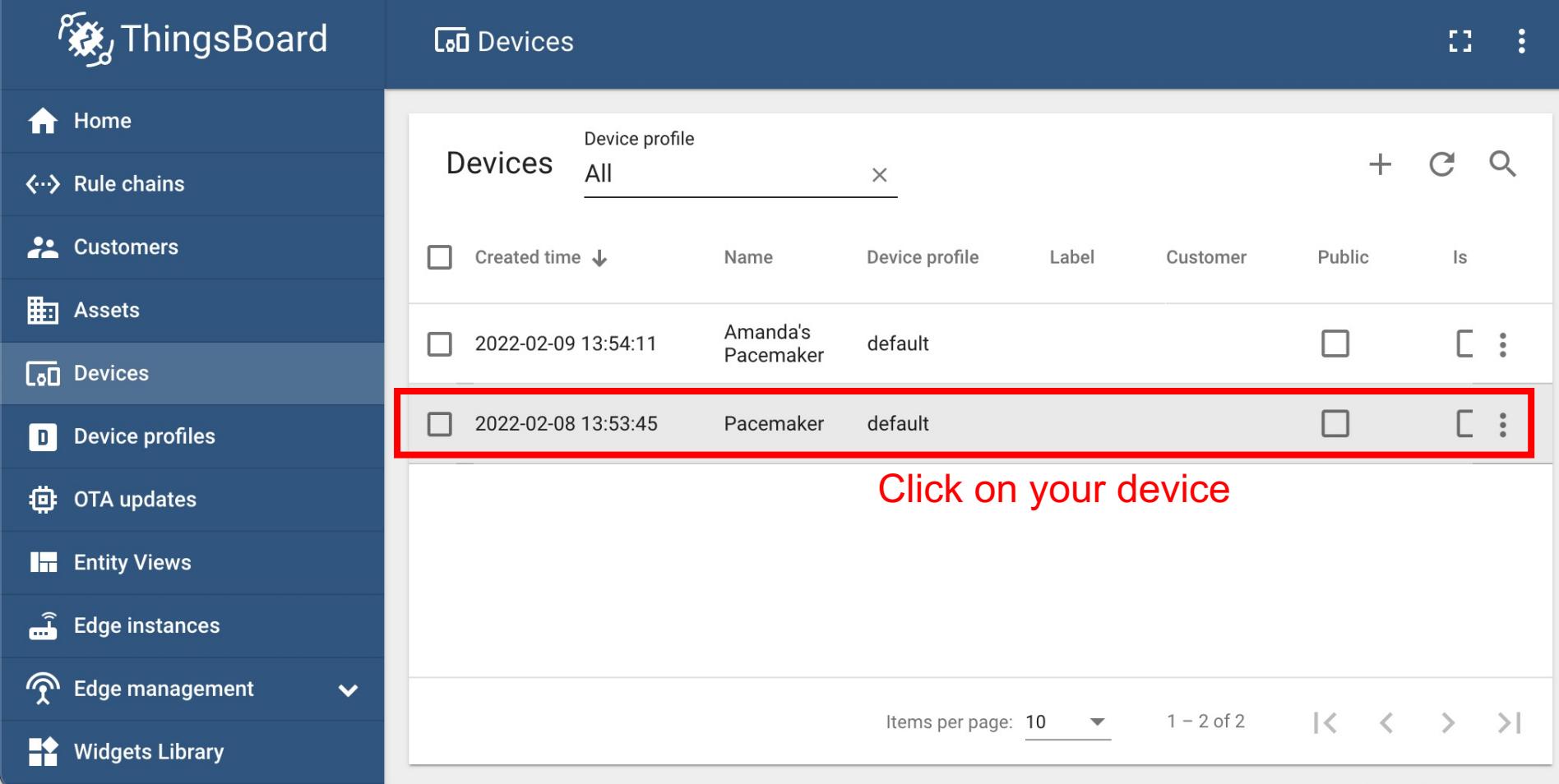
Create new device profile

[Next: Credentials](#)

[Cancel](#) [Add](#)

This screenshot shows the 'Add new device' dialog box. It has three tabs at the top: 'Device details' (selected), 'Credentials Optional', and 'Customer Optional'. The 'Name' field is required and contains 'Pacemaker'. The 'Label' field is optional and contains 'Add a Label Here for Legends Graphs etc'. Under 'Device profile', 'Select existing device profile' is selected with 'default' chosen. There are two radio buttons: one selected for 'Select existing device profile' and one for 'Create new device profile'. At the bottom right, there is a red box around the 'Add' button.

Device Creation

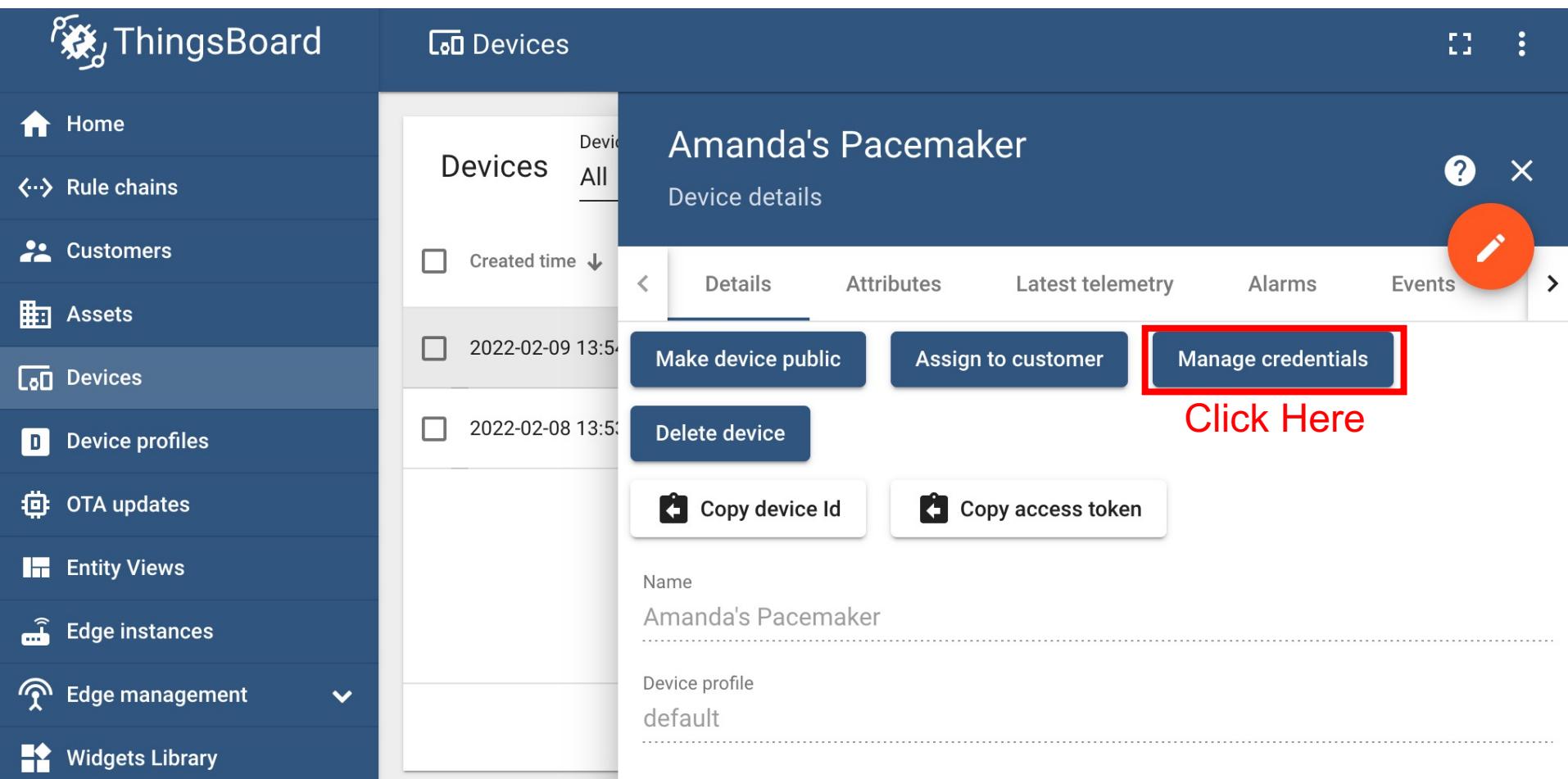


The screenshot shows the ThingsBoard web interface. The left sidebar has a dark blue background with white icons and text. The 'Devices' icon is highlighted with a light blue background. The main area has a dark blue header with the 'ThingsBoard' logo and a search bar. Below the header is a table titled 'Devices' with columns: Created time, Name, Device profile, Label, Customer, Public, and Is. Two rows are visible: one for 'Amanda's Pacemaker' and one for 'Pacemaker'. The row for 'Pacemaker' is highlighted with a red border. A red box surrounds the entire table area. Red text at the bottom center says 'Click on your device'.

Created time	Name	Device profile	Label	Customer	Public	Is
2022-02-09 13:54:11	Amanda's Pacemaker	default				
2022-02-08 13:53:45	Pacemaker	default				

Items per page: 10 | 1 - 2 of 2 | < < > >|

Device Creation



The screenshot shows the ThingsBoard web interface. The left sidebar has a 'Devices' icon selected. The main area shows a 'Devices' card for 'Amanda's Pacemaker'. A red box highlights the 'Manage credentials' button. A red arrow points to the 'Click Here' text.

ThingsBoard

Devices

Home

Rule chains

Customers

Assets

Devices

Device profiles

OTA updates

Entity Views

Edge instances

Edge management

Widgets Library

Devices

All

Created time ↓

2022-02-09 13:54:12

2022-02-08 13:53:42

Amanda's Pacemaker

Device details

Details Attributes Latest telemetry Alarms Events

Make device public Assign to customer Manage credentials

Delete device

Copy device Id Copy access token

Name: Amanda's Pacemaker

Device profile: default

Click Here

Device Creation

The screenshot shows the ThingsBoard interface with a modal dialog titled "Device Credentials". The dialog has a blue header and contains the following fields:

- Credentials type: Access token
- Access token *: YBbl38gA9WJ8a5qb5sYs (highlighted with a red box)

Below the form is a red text message: "Update to your device id". At the bottom are two buttons: "Cancel" and "Save" (highlighted with a red box).

The background shows the "Devices" section of ThingsBoard, with a device named "Amanda's Pacemaker" visible. The left sidebar includes links for Home, Rule chains, Customers, Assets, Devices (selected), Device profiles, OTA updates, Entity Views, Edge instances, Edge management, and Widgets Library.

Device Creation

The screenshot shows the ThingsBoard web interface. The left sidebar has a dark blue background with white icons and text: Home, Rule chains, Customers, Assets, Devices (selected), Device profiles, OTA updates, Entity Views, Edge instances, Edge management (with a dropdown arrow), and Widgets Library. The main area has a light blue header with the title 'Devices' and a sub-header 'Pacemaker Device details'. Below the header is a navigation bar with tabs: Details, Attributes, Latest telemetry (which is highlighted with a red border), Alarms, Events, and a search icon. A large orange circular button with a white pencil icon is located on the right side of the navigation bar. To the left of the main content, there is a sidebar titled 'Devices' with a 'Created time' dropdown menu showing two entries: '2022-02-09 13:54' and '2022-02-08 13:53'. The main content area displays 'Latest telemetry' data for the 'Pacemaker' device, showing a single entry: 'Last update time' (2022-02-09 14:02:28), 'Key' (data), and 'Value' ({"HR":21}). At the bottom of this section are buttons for 'Items per page' (set to 10), '1 - 1 of 1', and navigation arrows. The overall interface is clean and modern, typical of a cloud-based IoT platform.

If set up correctly, and data is streaming, you will see your device data here

Dashboard Creation

Dashboard Creation

ThingsBoard

Home Entity Views

Dashboard management

Widgets Library Dashboards

Audit

Audit Logs

Click Here

The screenshot shows the ThingsBoard web interface with the 'Entity Views' tab selected. The main content area is titled 'Dashboard management' and contains two buttons: 'Widgets Library' and 'Dashboards'. The 'Dashboards' button is highlighted with a red border and a red arrow points to it from the text 'Click Here' located below it. The left sidebar lists various navigation items: HOME, RULE CHAINS, CUSTOMERS, ASSETS, DEVICES, ENTITY VIEWS, WIDGETS LIBRARY, DASHBOARDS, and AUDIT LOGS.

Dashboard Creation

The screenshot shows the ThingsBoard web interface. The left sidebar contains navigation links: ThingsBoard, Devices, Device profiles, OTA updates, Entity Views, Edge instances, Edge management (with a dropdown arrow), Widgets Library, Dashboards (selected and highlighted in blue), Audit Logs, Api Usage, and System Settings. The URL in the address bar is <https://tb-exp.precise.seas.upenn.edu/dashboards>. The main content area is titled "Dashboards". It features a table with columns: Created time (sorted by descending time), Title, and Assigned to customers. A single row is visible, showing "2022-02-08 14:53:16" under "Created time", "Pacemaker" under "Title", and a "Public" status with a "Click Here" button and a red-bordered plus sign icon. The bottom of the page includes pagination controls (Items per page: 10, 1 - 1 of 1) and navigation arrows.

<input type="checkbox"/>	Created time ↓	Title	Assigned to customers
<input type="checkbox"/>	2022-02-08 14:53:16	Pacemaker	<input type="checkbox"/> Click Here Public

Items per page: 10 | 1 – 1 of 1 | < > >>

Dashboard Creation

The screenshot shows the ThingsBoard web application interface. On the left is a sidebar with various navigation items: Devices, Device profiles, OTA updates, Entity Views, Edge instances, Edge management (with a dropdown arrow), Widgets Library, Dashboards (which is the active tab, indicated by a blue background), Audit Logs, Api Usage, and System Settings (with a dropdown arrow). The main content area has a header "Dashboards". Below it is a table with two rows:

	Created time ↓	Title	Assigned
<input type="checkbox"/>	2022-02-08 14:53:16	Pacemaker	

To the right of the table is a context menu with two options: "Create new dashboard" (highlighted with a red box) and "Import dashboard". At the bottom right of the main content area, there is a red text overlay that says "Click Here".

Dashboard Creation

The screenshot shows the ThingsBoard web interface. On the left, there is a sidebar with various navigation options: Devices, Device profiles, OTA updates, Entity Views, Edge instances, Edge management, Widgets Library, Dashboards (which is currently selected), Audit Logs, API Usage, and System Settings. The main area is titled "Add Dashboard". It has fields for "Title *" (containing "Team Blue's Dashboard" with a red placeholder "Add Name Here" overlaid) and "Description". Below these is a section for "Mobile application settings". A "Dashboard image" field is present, showing a dashed box for dropping an image or selecting a file. The "No image selected" message is visible inside this field. At the bottom right of the dialog are "Cancel" and "Add" buttons, with "Add" being highlighted by a red rectangle.

Dashboard Creation

ThingsBoard

- Home
- Rule chains
- Customers
- Assets
- Devices
- Device profiles
- OTA updates
- Entity Views
- Edge instances
- Edge management
- Widgets Library
- Dashboards
- Audit Logs
- Api Usage
- System Settings

Dashboards > Pacemaker

Pacemaker

Pacemaker ▾ Pacemaker Realtime - last minute

Heart Rate

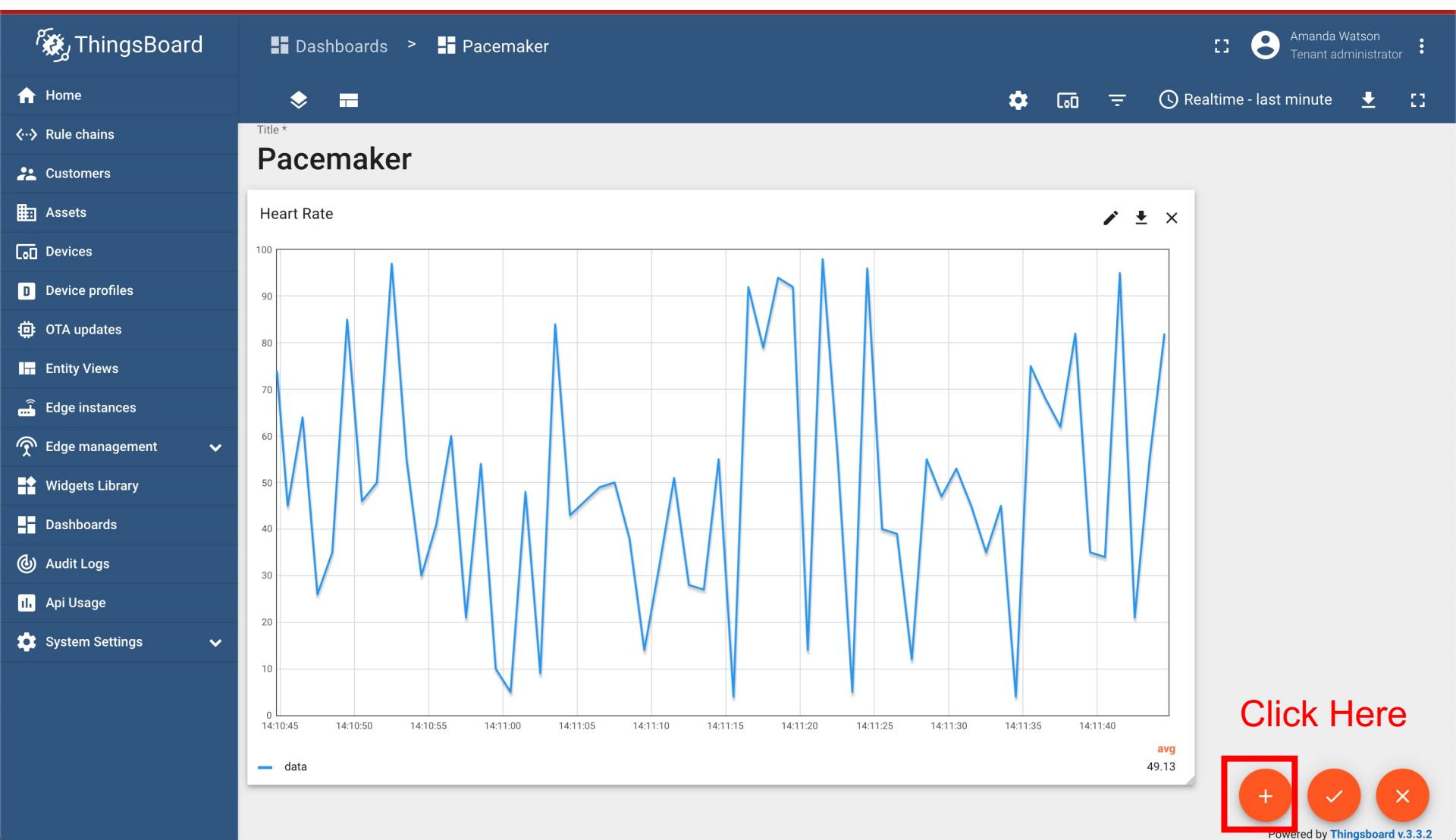
avg
NaN

data

Click Here

Powered by ThingsBoard v.3.0.2

Dashboard Creation



Click Here



Powered by [Thingsboard v.3.3.2](#)

Dashboard Creation

ThingsBoard

Dashboards > Pacemaker

Amanda Watson
Tenant administrator

Home Rule chains Customers Assets Devices Device profiles OTA updates Entity Views Edge instances Edge management Widgets Library Dashboards Audit Logs API Usage System Settings

Pacemaker

Heart Rate

Alarms

Type	Severity	Status
Temperature	Major	Cleared
Temperature	Critical	Cleared
Low Humidity	Warning	Active
Low Humidity	Warning	Active

Alarm widgets

System

Visualization of alarms for devices, assets and other entities.

Analogue gauges

System

Display temperature, humidity, speed, and other latest values on various analog gauge widgets.

Cards

System

Tables and cards to display latest and historical values for multiple entities simultaneously.

Charts

System

Display timeseries data using customizable line and bar charts. Use various pie charts to display latest values.

Control widgets

System

Send commands to devices.

Date

System

Contains widgets to change the data range for other widgets on the dashboard.

Digital gauges

System

Edge #1 Quick Overview

Assigned to: Customer A

Assets

Edge widgets

System

Dashboard Creation

ThingsBoard

- Home
- Rule chains
- Customers
- Assets
- Devices
- Device profiles
- OTA updates
- Entity Views
- Edge instances
- Edge management
- Widgets Library
- Dashboards
- Audit Logs
- Api Usage
- System Settings

Dashboards > Pacemaker

Pacemaker

Pacemaker ▾ Pacemaker Realtime - last minute

Heart Rate

avg
NaN

data

Amanda Watson
Tenant administrator

Powered by [Thingsboard v.3.3.2](#)

Dashboard Creation

The screenshot shows the ThingsBoard interface for creating a dashboard. The main title is "Pacemaker". On the left, there's a sidebar with various navigation options like Home, Rule chains, Customers, Assets, Devices, Device profiles, OTA updates, Entity Views, Edge instances, Edge management, Widgets Library, Dashboards, Audit Logs, Api Usage, and System Settings.

The central area displays a line chart titled "Heart Rate" showing fluctuating data over time. A modal window titled "Data key configuration" is open, allowing customization of the data key. The "Settings" tab is selected, showing:

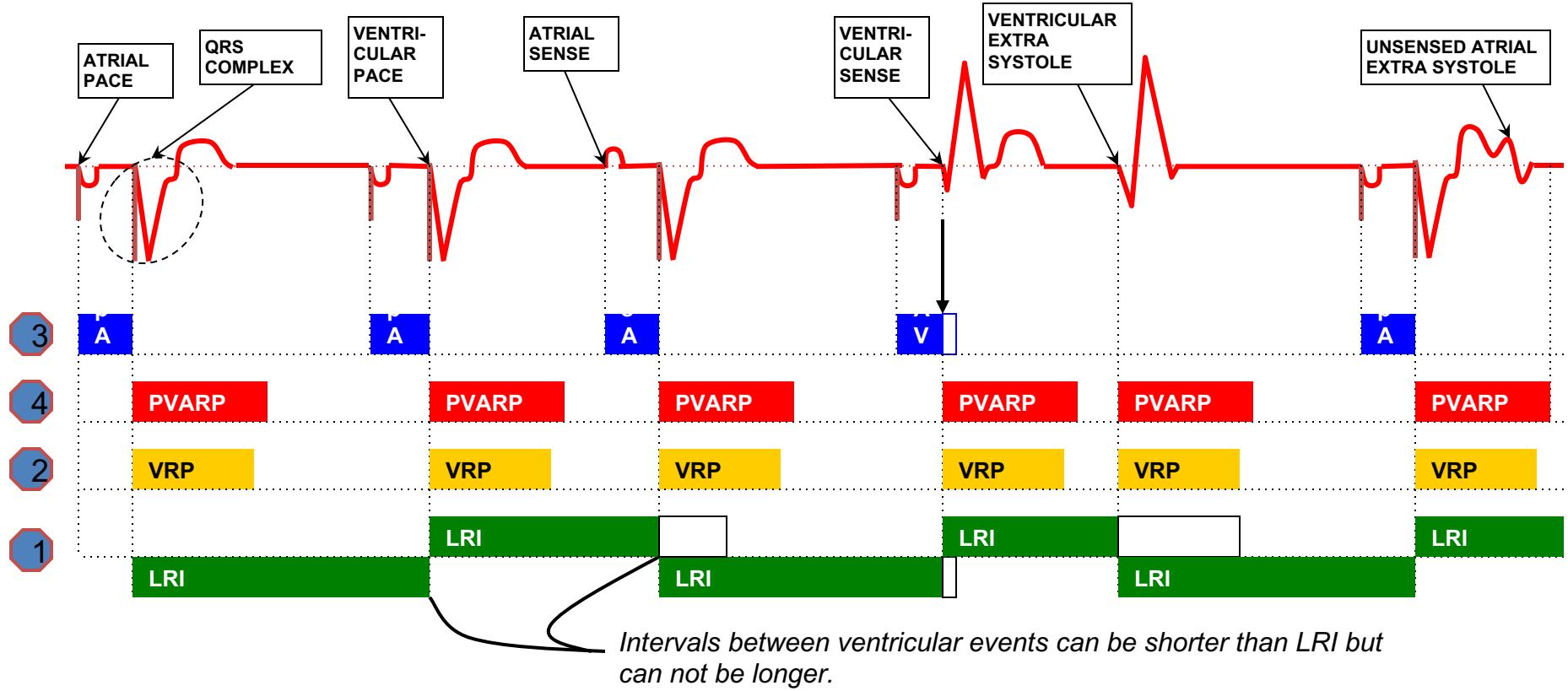
- Key ***: data
- Label ***: data
- Color ***: #2196f3 (blue)
- Special symbol to show next to value**: None
- Number of digits after floating point**: None
- Use data post-processing function**: checked
- function** (time, value, prevValue, timePrev, prevOrigValue) {
1 return parseInt(JSON.parse(value).HR)
2 };
3 }

At the bottom of the modal are "Cancel" and "Save" buttons.

Pacemaker Appendix:

More on realistic pacemaker timing constraints

Four Fundamental Timing Cycles

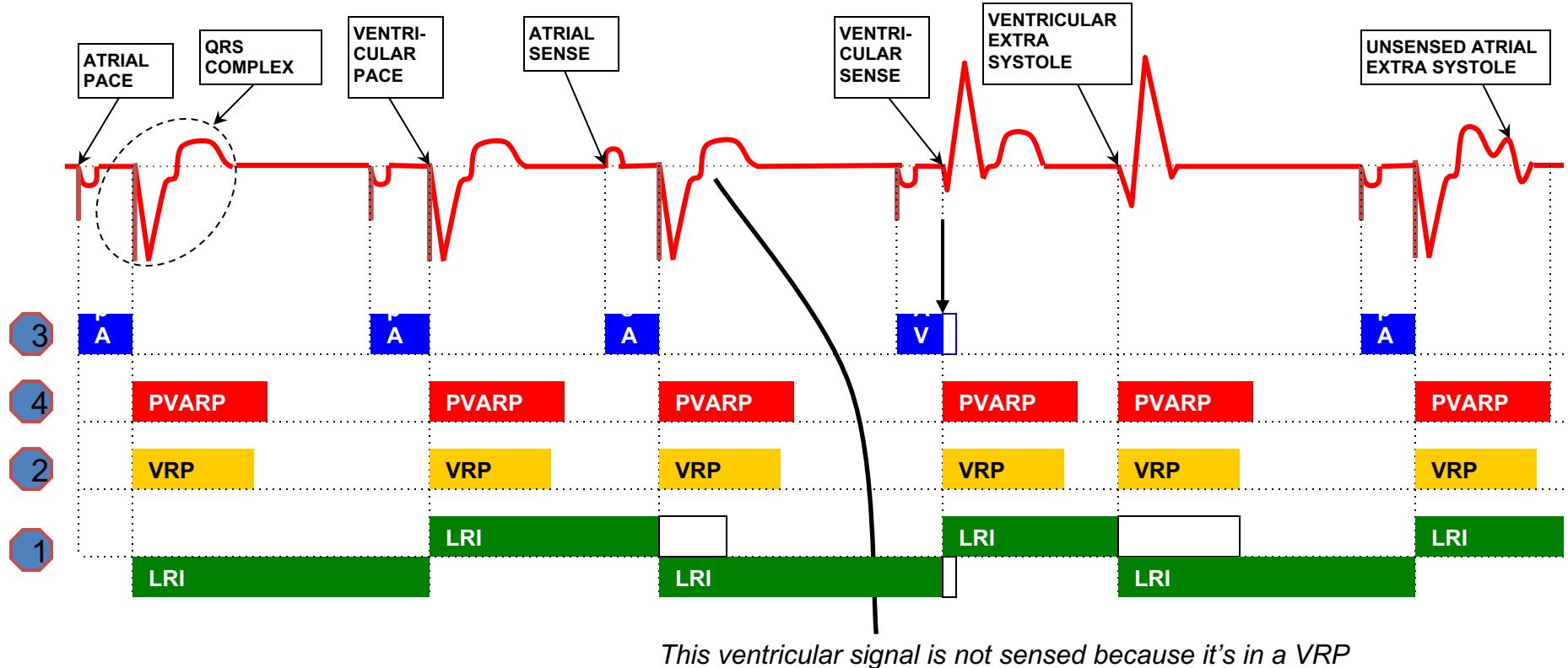


1 LRI = Lower Rate Interval

Longest interval between a paced or sensed ventricular event and the succeeding ventricular paced event with out intervening sensed events.

That is, the lowest allowable rate of ventricular events for normal operation of the heart.

Four Fundamental Timing Cycles

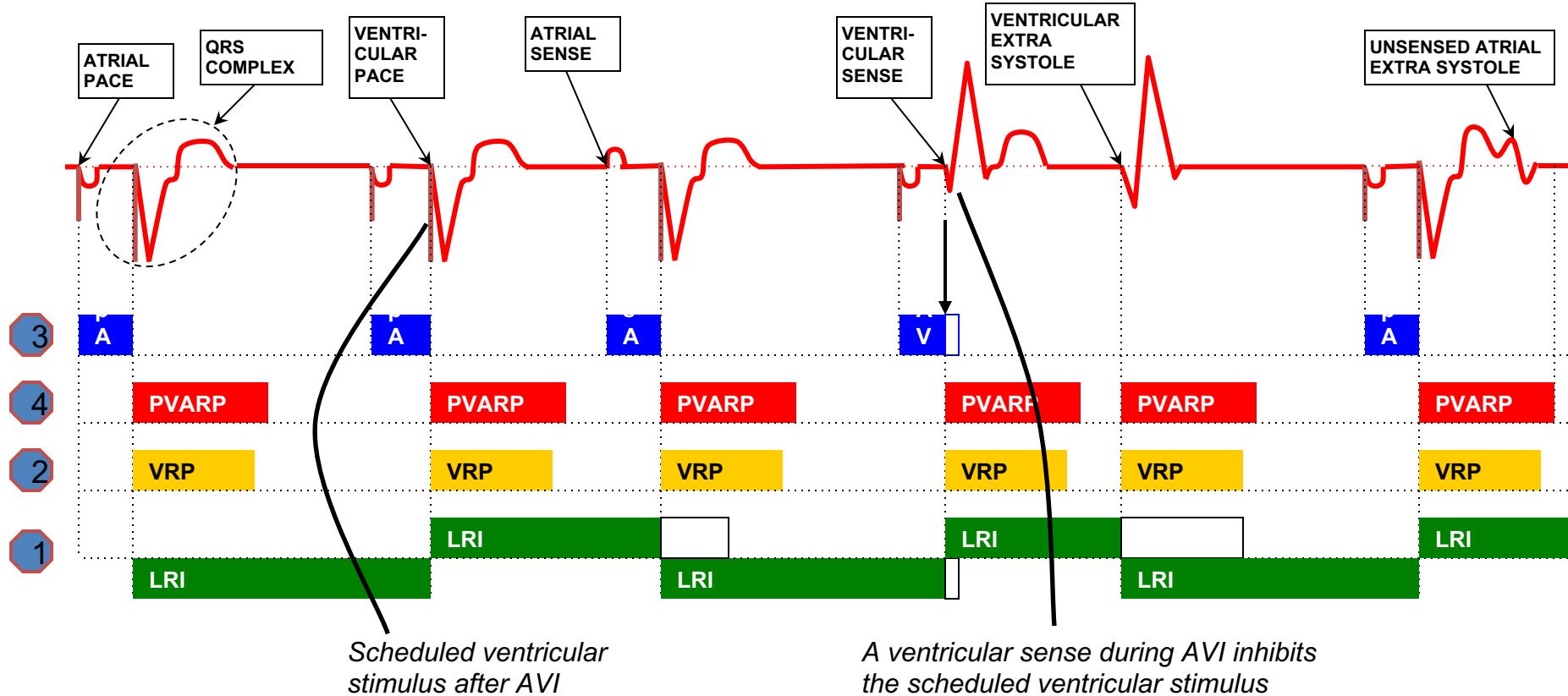


2 VRP = Ventricular Refractory Period

Interval initiated by a ventricular event during which a new LRI cannot be initiated.

After a ventricular event, there are signals (own stimulus, QRS complex, after potential, ...) which can be identified incorrectly as ventricular events, thus initiate a new LRI. VRP is used to avoid this.

Four Fundamental Timing Cycles



3 AVI = AtrioVentricular Interval

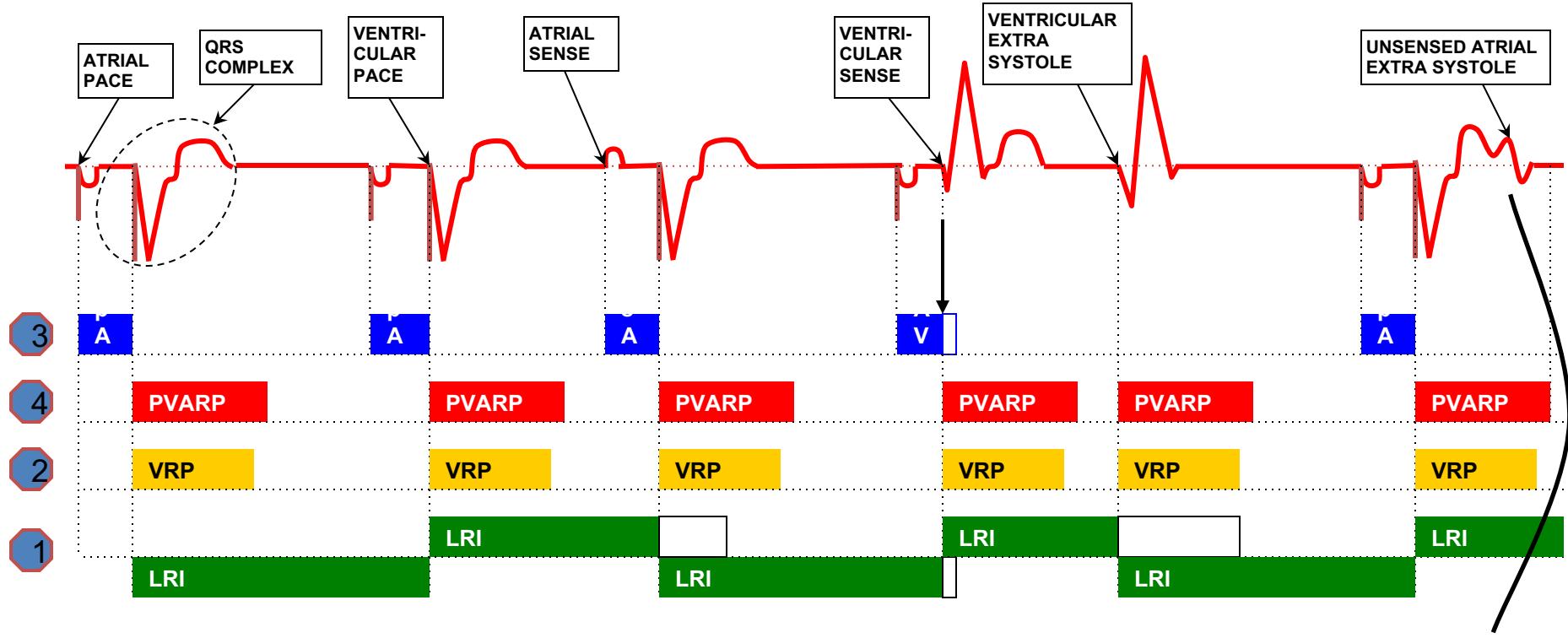
Interval between an atrial event and the scheduled delivery of a ventricular stimulus.

In a normal heart, an atrial event must always be followed by a ventricular event after some delay (AVI) \Rightarrow AV synchrony.

pAVI for paced atrial events; sAVI for sensed atrial events.

Maybe fixed or rate-adaptive.

Four Fundamental Timing Cycles



This atrial event is not sensed because it's in PVARP; no AVI is initiated

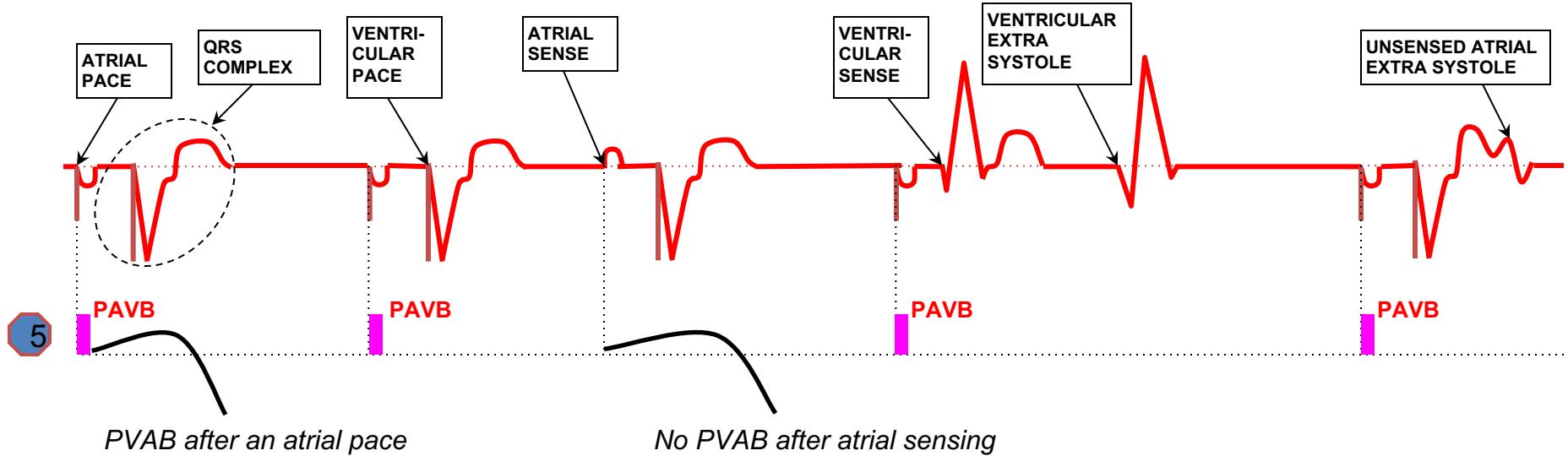
4

PVARP = PostVentricular-Atrial Refractory Period

Interval after a ventricular paced or sensed event during which an atrial event cannot initiate a new AVI.

To prevent the atrial channel from inappropriately sensing ventricular events (QRS complex, ventricular stimuli,...) or retrogradely P waves.

Fifth Timing Cycle to Prevent AV Crosstalk



AV Crosstalk

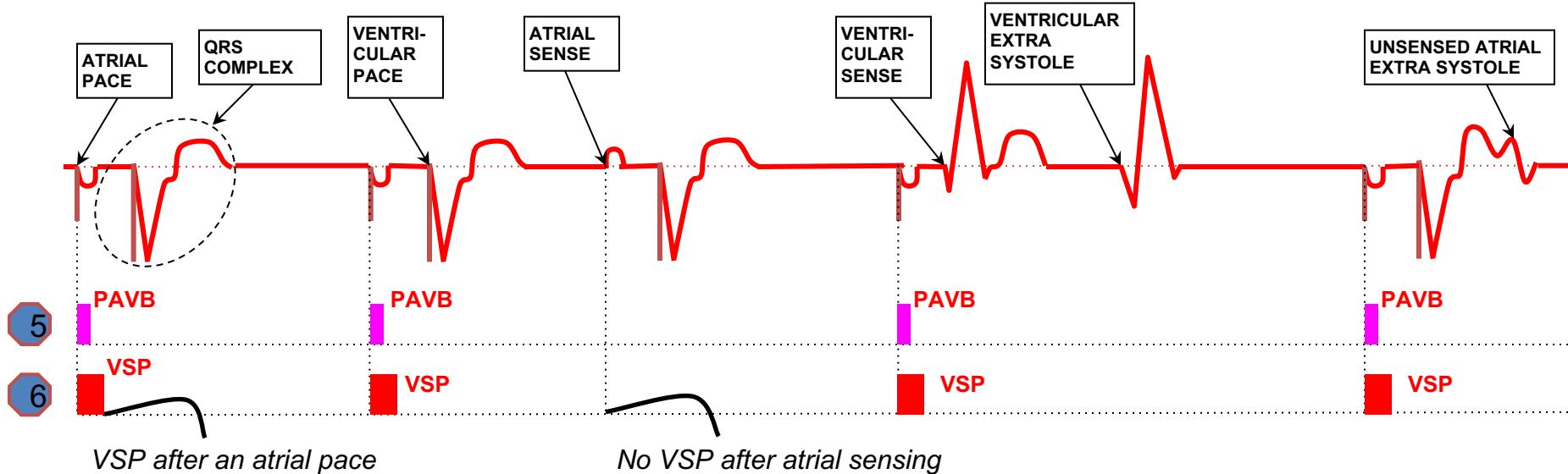
The disturbance caused by an atrial stimulus which, if sensed by the ventricular channel, may cause ventricular inhibition.

5 PAVB = Post-Atrial Ventricular Blanking

Brief interval (10-60ms) initiated by an atrial output pulse when the ventricular channel is switched off and cannot sense.

There is no PAVB after an atrial sense since it does not cause disturbance.

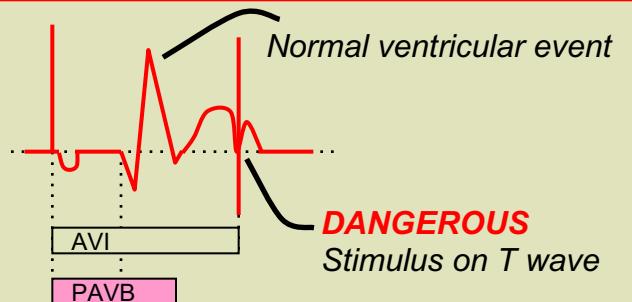
Sixth Timing Cycle to Prevent the Consequences of AV Crosstalk

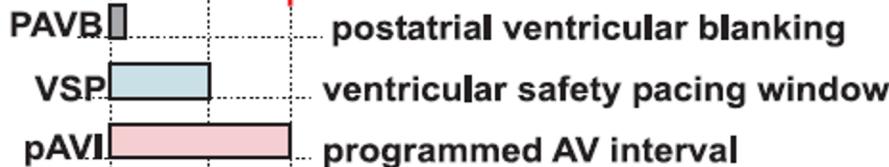
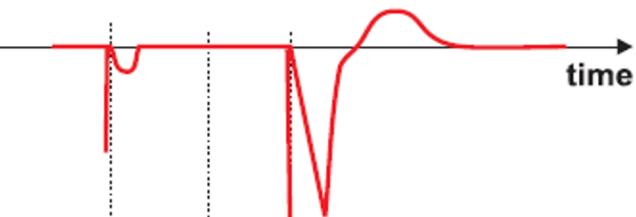


6 VSP = Ventricular Safety Pacing

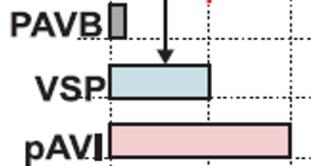
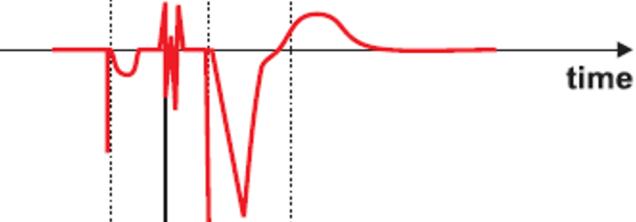
First part of AVI (PAVB < VSP < AVI) during which ventricular channel can sense; a signal sensed in VSP but not in PAVB will trigger a premature ventricular stimulus at the end of VSP (thus shorten the current AVI).

VSP does not prevent crosstalk, just prevents its consequences.

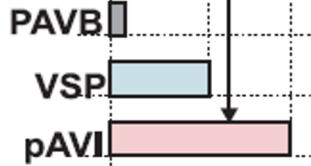
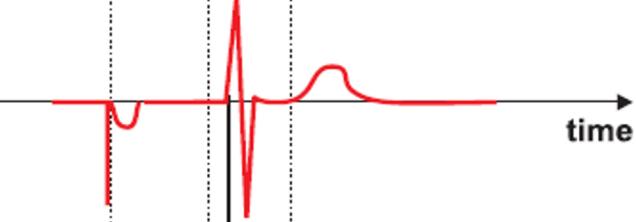




No spontaneous conduction, no crosstalk, no interference : stimulation at the end of the programmed AV interval



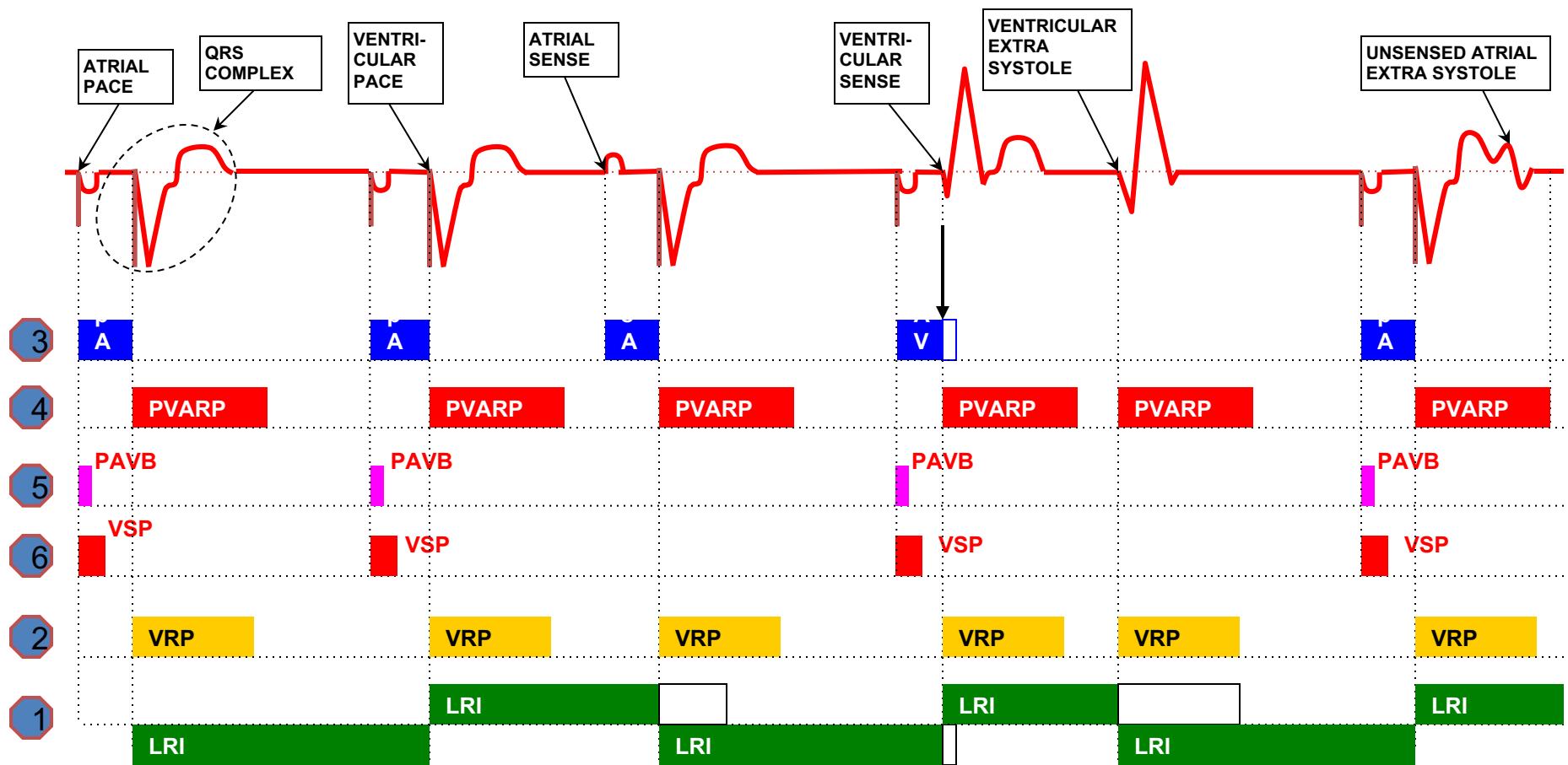
Interference (or early QRS) during the VSP window (beyond the PAVB) results in a committed ventricular stimulus at the end of that window and a characteristic shortening of the AV interval



Normal inhibition of the ventricular channel by a conducted QRS

Intrinsic P-R intervals are usually longer than 100 to 110 ms, therefore the VSP window is often called a *non-physiologic AV delay*

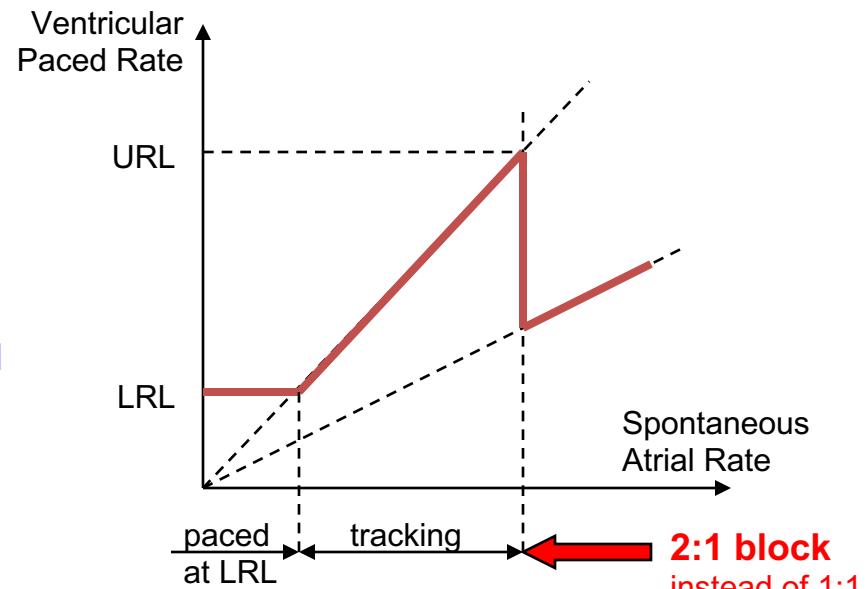
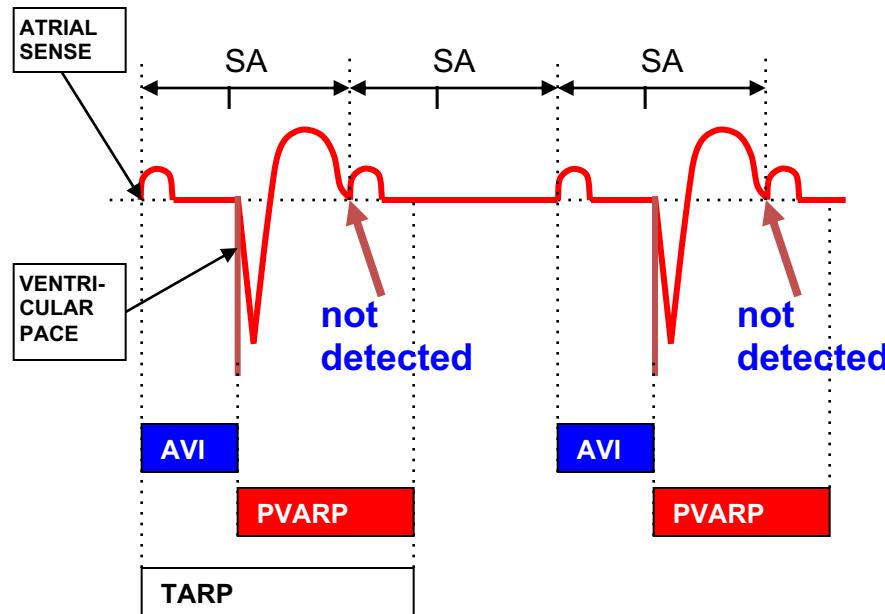
The TARP Timing Cycle



$$\text{TARP} = \text{AVI} + \text{PVARP} = \text{Total Atrial Refractory Period}$$

Tracking Mode

The ventricular paced rate follows the spontaneous atrial rate (1:1).

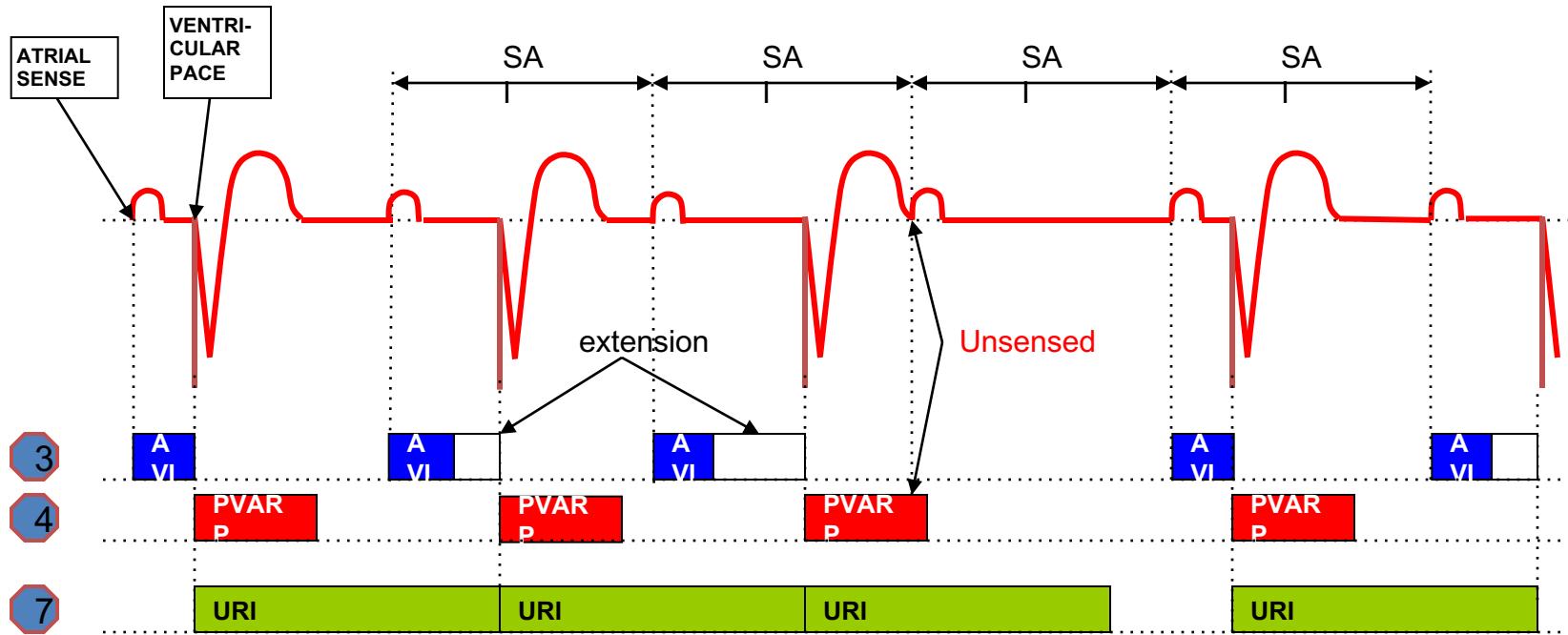


SAI = Spontaneous Atrial Interval

$$\text{URL} = 60,000 / \text{TARP}$$
$$\text{LRL} = 60,000 / \text{LRI}$$

Upper rate limitation by the abrupt development of 2:1 block should be prevented!

Seventh Timing Cycle to Avoid 2:1 Block



7

URI = Upper Rate Interval (programmable)

The shortest interval between consecutive ventricular paces.

If SAI < URI, AVI may be lengthened.

URI must be longer than TARP (AVI + PVARP), otherwise it has no effect.