Developer's Guide To Pacemaker Development

Tutorial 2: Pacemaker Functionality

SFWRENG/MECHTRON 3K04
McMaster University

Michael Kehinde
kehindem@mcmaster.ca

November 5, 2020

Pacing, Sensing and Timing Cycles

The aim of this tutorial is to describe pacemaker nomenclature and the basics of pacing and sensing. This tutorial introduces timing cycles — a useful approach for depicting pacemaker functionality and interactions with the heart while showing the relationships between pacemaker programmable parameters and cardiac events. A thorough understanding of the concepts discussed in this tutorial will build your confidence with decomposing the requirements and designing a FSM model. By the end of this tutorial you will be able to generate paces in either the atrium or ventricle by creating a Stateflow chart that refers to the switches in the atrial and ventricle circuit diagram. You will also be able to implement a pacemaker operating mode.

Topics Covered

- What are pacemaker modes?
- What is PWM, how does it work and why do we use it?
- Pacemaker sensing concepts: undersensing, oversensing and sensing threshold
- Basic pacemaker timing parameters: pacing period and pulse width
- Introduction to timing cycles

Prerequisites Read **Pacemaker Shield Explained**

Reference Materials Pacing Modes
Timing Cycles
Advanced Timing Cycles

1 Background

A pacemaker performs two primary functions: **1) pacing** and **2) sensing**. **Pacemaker modes** determine which functions are to be performed by the pacemaker, and in which chambers those functions are to be performed. For a specified mode of operation, the fundamental functionality of the pacemaker is to perform a primary function at the right time.

As such, two kinds of logic are employed when implementing software for a pacemaker:

1. **timing logic:** entails utilizing timing mechanisms to determine *when* to perform a primary function (and when *not* to perform a primary function)
2. **circuitry logic:** entails reading or writing appropriate values to the GPIO pins in order to perform the primary function

The behaviour of the pacemaker under a specified mode is depicted with **pacemaker timing cycles**.

1.1 Pacemaker Operating Modes

Pacemaker operating modes communicate the capabilities and programming of pacemakers, and are described using pacing codes. The NBG Cardiac Pacing Code is an international standard letter code that identifies the chambers that are paced and sensed, and how the pacemaker will respond to intrinsic events. The NBG Cardiac Pacing Code is described in **Section 3.5 — Bradycardia Operating Modes** of the **PACEMAKER System Requirements Specification**.

	I	II	III	IV (optional)
Category	**Chambers Paced**	**Chambers Sensed**	**Response To Sensing**	**Rate Modulation**
Letters	O–None A–Atrium V–Ventricle D–Dual	O–None A–Atrium V–Ventricle D–Dual	O–None T–Triggered I–Inhibited D–Tracked	R–Rate Modulation

Figure 1: Bradycardia Operating Modes (from PACEMAKER SRS)

Collating the information in the figure above results in many different modes of pacing. The following table outlines the most common pacemaker modes and a brief description of each mode. The explanation of the behaviour of the pacemaker under the specified mode is left as an exercise.

Code	Description	Explanation
AOO	Atrial asynchronous pacing	
VOO	Ventricle asynchronous pacing	
AAI	Atrial demand pacing	
VVI	Ventricle demand pacing	
DOO	Dual asynchronous pacing	
DDDR	Dual rate adaptive demand pacing	

Table 1: Explanation of Pacemaker Modes

In practise, a physician would choose a reasonable pacemaker mode based on a number of factors such as the cardiac need of the patient. For a pacemaker in Permanent State, the physician would command a mode switch by first changing the **“Mode”** programmable parameter to an appropriate value. For more information on pacing mode selection, refer to the set of slides **Pacing Modes**.

1.2 Pacemaker Functionality

Pacing Concepts

A pacemaker performs a pacing function by generating an output stimulus. Section 3 of the document **Pacemaker Shield Explained** describes how to operate the pacemaker hardware reference platform to generate pacemaker output stimuli. Section 3.3 of the document **Intro To HeartView** illustrates the desired waveform of the pacing output.

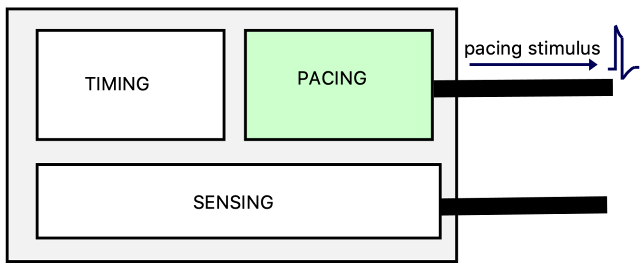
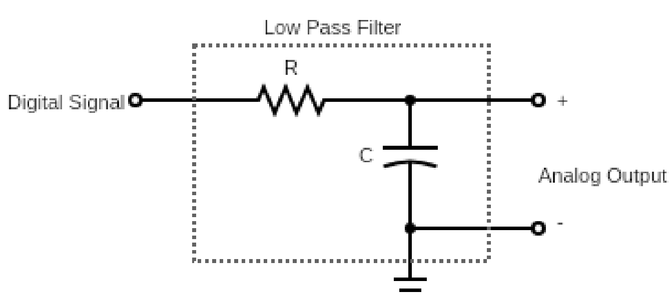


Figure 2: Pacemaker Functional Decomposition — Pacing

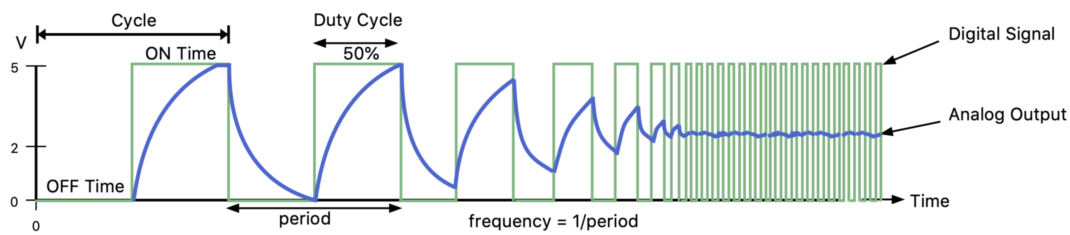
Pulse Width Modulation (PWM)

PWM is a power-efficient, dynamically responsive and cost effective approach for producing an analog voltage from a digital system.

A frequently enquired topic is the notion of *how* the PWM technique is employed by the pacemaker shield to allow us to control voltage levels in the circuit. The following figures aim to illustrate the concept. On the left side of Figure 3a is an input representing a digital signal that has two increments: ON/OFF (or HIGH/LOW). On the right side is an analog voltage that has an infinite number of increments. In between is a passive low pass filter with time constant RC and corner frequency f_c. The frequency response of a LPF indicates that the LPF "smoothes" the input signal by removing its high frequency components at frequencies beyond f_c.



(a) LPF Circuit

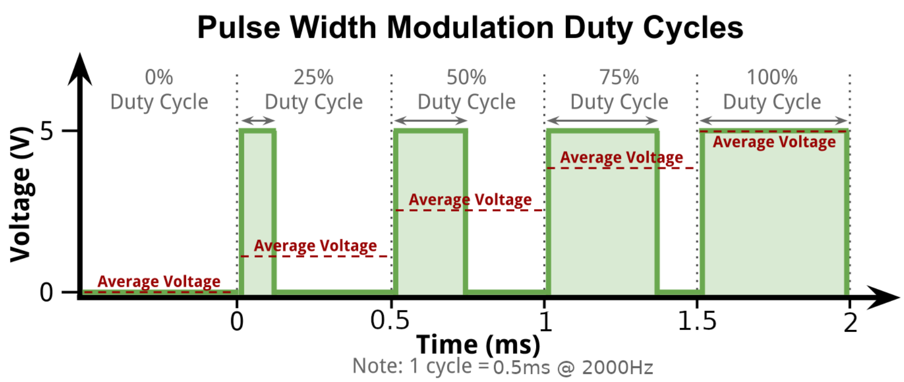


(b) Digital Signal Experiment

Figure 3: Digital Signal Experiment with LPF

Consider the experiment illustrated in Figure 3b. Initially, the digital signal is pulsed on and off at a low frequency, and its frequency is gradually increased. Within each cycle, the signal is on for 50% of the time, and off for the other 50%. When the signal pulses ON, charge builds in the capacitor; when the signal pulses OFF, the capacitor discharges. At sufficiently high frequencies, the output attenuates to a smooth voltage that is linearly proportional to the duty cycle.

A PWM signal is a special kind of digital signal that has a fixed frequency and variable duty cycle. A reasonable frequency would be chosen based on the characteristics of the LPF circuit such that the output signal is smooth; however, the duty cycle remains adjustable so as to permit varying the output voltage level. It can be said that the LPF performs an "averaging" function on the PWM signal (*i.e. an input voltage that is 5V for 50% of the time and 0V for another 50% percent of the time results in an average voltage of 2.5V*).



The LPF represents Component 3 on Figure 7 (Pacing Flowchart Overview) in **Pacemaker Shield Explained**. For the pacing circuitry, the output voltage is used to charge capacitor C22. Refer to Figure 12 in **Pacemaker Shield Explained**: by varying the output voltage, what pulse characteristic of the pacing stimulus would be affected? (Note that the pulse width of the PWM signal is *not* the same as the pace width of the pacing stimulus!).

Sensing Concepts

The document **Pacemaker Shield Explained** describes how to control the sensing circuitry on the pacemaker hardware reference platform to sense natural heart signals.

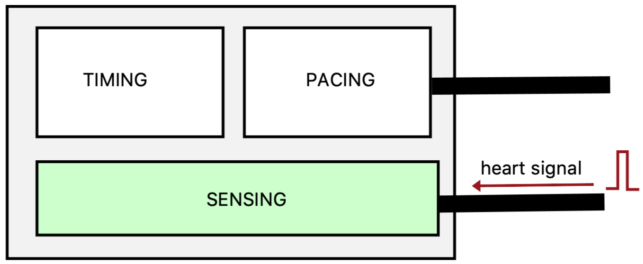
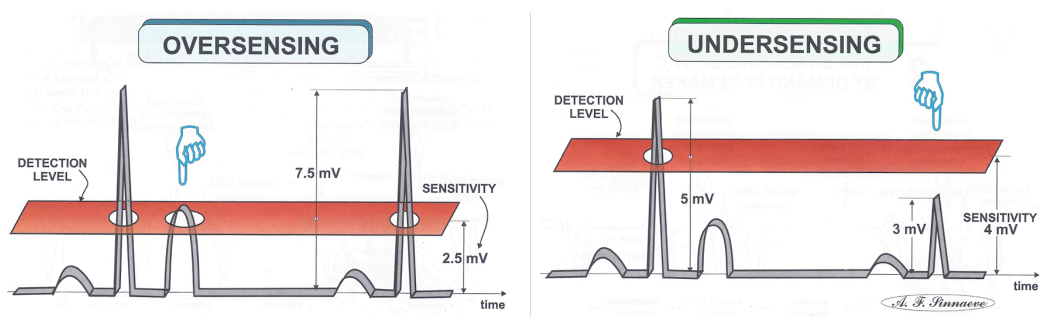


Figure 4: Pacemaker Functional Decomposition — Sensing

A key concept for programming a pacemaker is the determination of sensitivities (also known as *sensing thresholds*). Sensitivity refers to a programmable parameter of the pacemaker and is controlled in the sensing circuit by setting the duty cycle for the sensing threshold signals using the appropriate PWM pins.

The voltage in the following figure refers to the intracardiac electrogram (*not* the surface QRS complex). A sensitivity of 4 mV means that the pacemaker can only sense a signal equal to or greater than 4 mV. It will sense a signal of 5 mV but not a signal of 3 mV.



In practice, a reasonable sensitivity level would be chosen to avoid undersensing (i.e. incorrectly failing to sense events) and oversensing (i.e. unintended sensing of certain events).

A simple exercise for determining reasonable levels for the sensing threshold is to pass the natural heart signals from HeartView to your pacemaker and probe the signal on a Scope in real time (using Monitor and Tune) by reading the ATR_RECT_SIGNAL or VENT_RECT_SIGNAL analog input pins. (Remember to set the appropriate pin to enable the sensing circuit first!) These signals represent the voltage seen by the comparator (Component 8 of the Sensing Flowchart) that is compared to the sensing threshold to produce the ATR_CMP_DETECT and VENT_CMP_DETECT signals respectively.

Once a reasonable threshold has been set, sensing can be performed by simply reading the appropriate detect pins (see Pacemaker Shield Explained). Note that the shape and magnitude of the HeartView waveform is different from the waveform shown above. The waveform will also appear differently from what is shown on the HeartView, since it will have been processed by the pacemaker shield.

Timing

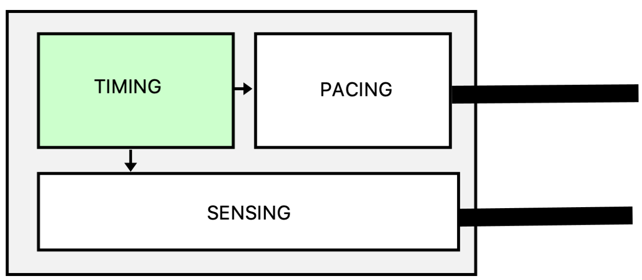


Figure 5: Pacemaker Functional Decomposition — Timing

The timing functionality of a pacemaker is to coordinate when to perform pacing and sensing (as well as when *not* to pace or sense). Timing can be achieved by using timing delays to track time intervals. Timing delays can be implemented in a Stateflow chart with the use of temporal transition labels. The code generation process converts temporal transition labels into instructions that utilize the timer on the MCU.

1.3 Pacemaker Timing Cycles

Pacemakers function based on timing cycles. Timing cycles are shown on ECG-like schematic diagrams and are used to represent the functionality of the pacemaker under a specified mode. A timing cycles graph is a useful tool for identifying relationships between sensed and paced cardiac events over time.

The following figure is an example of basic timing cycles for VVI that was adapted from the Timing Cycles slide deck. The diagram was modified for simplicity purposes, so that the voltage in the figure reflects a polarity that is referenced to the tip electrode (similar to Lead V6 ECG) and so that the pacing spikes will appear upwards (rather than downwards).

The diagram is made up of two parts: a timing cycles plot and annotations below the plot. The horizontal axis of timing cycles plot represents time; the vertical axis represents voltage. Below the plot are annotations for the cardiac event markers and time-locked labels to denote timing intervals. Timing intervals are either time-based programmable parameters, or values that can be determined by taking the inverse of a rate-based programmable parameter.

Recall from the beginning of this document that programming a pacemaker is essentially like solving a timing problem to determine when to pace/sense and when not to pace/sense. The decision of when to pace and sense depends on the mode and the meaning of the timing intervals.

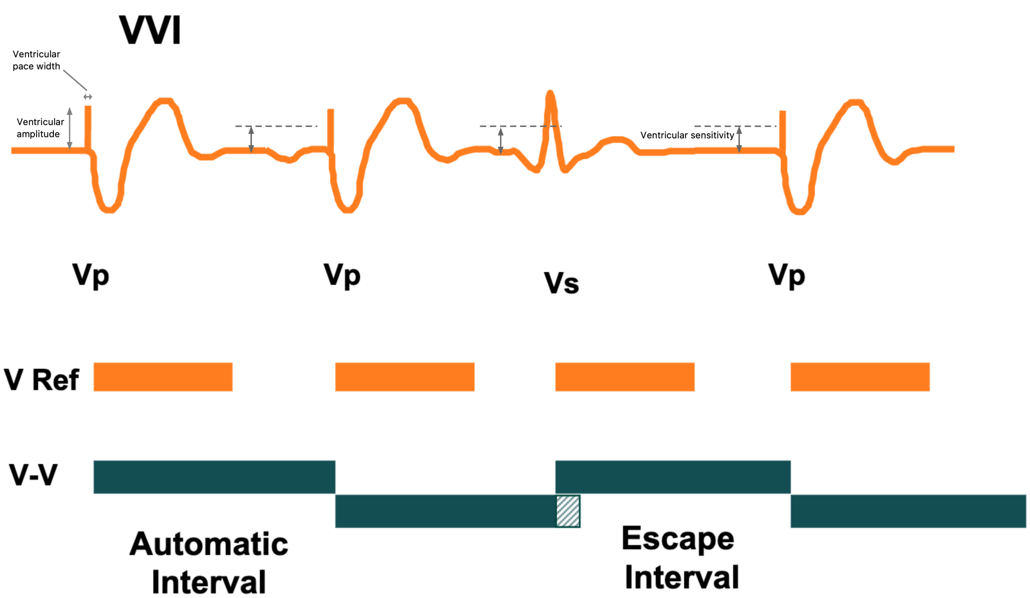


Figure 6: VVI Timing Cycles (adapted from: Timing Cycles)

A paced pulse is annotated with the Vp event marker. The first "V" in "VVI" means that the pacemaker plans to automatically pace the ventricle once every timing cycle. An automatic interval is the duration of a timing cycle and is referenced from the beginning of Vp. In other words, ventricular pacing is performed at the beginning of each pacing period. The pacemaker intends to release the next pacing stimulus once the automatic interval expires, which will initiate a new timing cycle.

The automatic interval is one of the programmable parameters for VVI. Refer to Section 5 of the PACEMAKER SRS: which parameter is equivalent to the automatic interval?

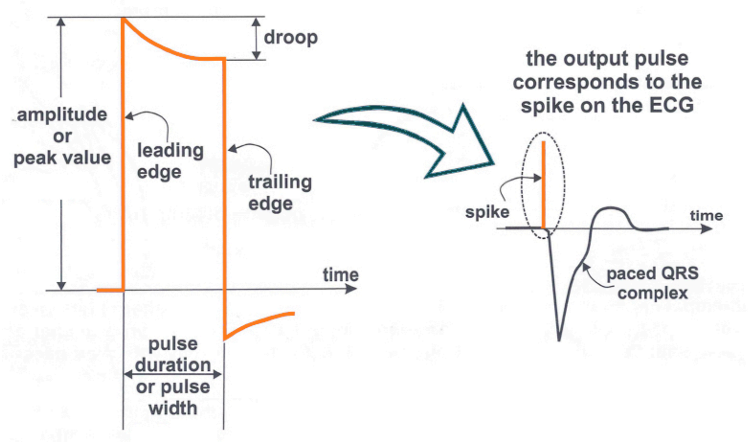


Figure 7: The Output Pulse of the Pacemaker

The spikes on the plot are pacemaker output pulses. The pacemaker stimulus charges the heart to a large voltage for *pulse duration* milliseconds; however, the pacing process does not terminate with the trailing edge. After the trailing edge, the large voltage is slowly dissipated over a much longer period than the brief pacemaker stimulus. Following the pulse is a paced QRS complex — the natural response of heart as observed from an ECG. (Reminder: the HeartView test suite is not intended to behave as a conforming heart. Refer to the note on Page 16 on Tutorial 1.4: Natural Heart Signals.)

The second "V" in "VVI" means that the pacemaker will perform sensing in the ventricle. However, to avoid sensing the artifacts of the paced QRS complex for the same heartbeat that was produced by the pacemaker, the pacemaker does not respond to sensing within the refractory period. The desired behaviour of the pacemaker is to sense the ventricle signal for a new heartbeat. As a result, the pacemaker must only respond to sensing when a sensed ventricular signal is detected outside the refractory period (V Ref above). A sensed cardiac event occurs when the natural ventricular signal exceeds the ventricular sensitivity and is denoted by the Vs event marker.

The third letter in "VVI" indicates that the pacemaker will respond to sensing by inhibiting the pace that it had planned to deliver. As a result, the pacemaker will reset its timer by commencing a new pacing period that begins at the onset of the sensed ventricular activity. The new pacing period is called the escape interval and its duration will be the same as the automatic interval.

Additional Comments

Figure 6 makes use of only 5 programmable parameters. Incorporating the remaining programmable parameters into the timing diagram (and/or creating new timing diagrams to understand the remaining parameters) is left as an exercise.

For more examples on timing cycles, refer to the reference materials listed on Page 2 of this document. For more on VVI timing cycles, see Schematic diagram of the pacemaker timing cycles during VVI pacing.

General Process for Using Timing Cycles Diagrams

1. Identify meaningful and applicable programmable parameters for the mode.
2. Assess the meaning of the mode and its programmable parameters to determine when (and when not to) perform pacing or sensing.
3. Identify different conditions for sensed events where the pacemaker would behave differently with response to monitored variables.
4. Plot the functionality of the pacemaker under each condition. Annotate the diagram using meaningful pulse and rate parameters.
5. Decompose the timing diagrams into segments that have distinct behaviour with respect to your controlled variables. Think in terms of states and transitions. Identify logical decisions by collating the timing diagrams to determine out where different decisions are made.
6. Determine what actions should be performed within each state.
7. Translate the diagram into a Stateflow implementation. Use the time intervals to inform your timing logic. Use the decisions to inform potential conditions between states.

2 Tutorial

1. At any given time, the pacemaker can only be in one state at a time. This means that the behaviour of a pacemaker under a specified mode can be represented as a finite state machine (FSM). What are advantages of using FSM to model the pacemaker?

2. Refer to Figure 7 in this document. Can you draw a single vertical line on the left image in Figure 7 to distinguish the two states shown in Figure 9 of pacemaker shield explained?

3. Similarly, the timing cycles above can further be decomposed into distinct states. Can you derive additional states by decomposing Figure 6? Draw vertical lines to identify the time when the pacemaker would enter a new state.

4. Design and implement the Stateflow charts for Assignment 2. For each chart,

   (a) Refer to Table 6 of the requirements for a list of meaningful parameters. Determine the mode you are trying to implement and which parameters are meaningful. If you find that a parameter is not applicable for a mode, you may exclude the parameter from your implementation and provide reasonable justification in your documentation.

   (b) Draw a timing diagram to help you make sense of the relevant parameters. For more complex modes, it may be worthwhile to draw several timing diagrams to describing different conditions for the same mode and represent different decisions in your software.

   (c) Design the Chart interfaces. Use appropriate data types to represent the data across your chart: inputs, outputs, parameters, enumerations, etc. Remember the golden principle: information hiding! Think about what kinds of changes your model should be able to accommodate to determine which variables should be exposed to the interface (and which should be internal to the chart).

   (d) Implement your timing and circuitry logic in the Chart. (Use Figure 9 in pacemaker shield explained as a reference for creating the states for VOO or AOO. More complex modes expound upon these two).