Software Development - 3K04

Assignment 3 - Part 2 - DCM

Instructor: Dr. Alan Wassyng

December 11, 2017

Group 7 - "Group 4"

Adam Cool	400032857	coola
Andrew Rehkopf	001412499	rehkopaz
Evan Gagich	400027794	gagiche
Ibrahim Tannir	001204850	tanniri
Michael Henry	400011728	henrym1
Michael Soosaipillai	400034820	soosaipm
Moshiur Howlader	001316948	howlam

Contents

1 Introduction

The DCM from assignment 2 was improved in a variety of ways. The most visible change has been made to the control interface where the electrogram window has been merged and connection to the pacemakers has been simplified. Specific details of the implementation pertaining to the login, new user and parameter storage can be referenced in previous documentation as these modules have not changed for assignment 3.

2 Control Interface

2.1 Design Decisions

The available serial ports is now displayed on the control interface to allow the user to connect and disconnect without manually changing which port is to be connected to. In addition, a GUI lamp has been added to indicate whether the user is connected or disconnected to a pacemaker.

The user may now control whether they are or are not connected to the pacemaker while before, they had no control over the connection. Granting the user control may seem like a improper use of hardware hiding however we believe that this additional control and data access allows the user to understand how exactly they are interfacing the pacemaker in much greater detail.

A private global variable has been added to act as a flag indicating whether the pacemaker is connected or not. The simple flag is used to prevent connecting to null ports. There are no associated functions with the control interface however upon connection to the pacemaker, a MATLAB serial object is initialized and its send and receive functions are used continuously.

2.2 Future Changes

The requirements of this interface would be subject to change in later designs as this screen controls the pacemaker interface and if there are changes made to the pacemaker, this interface must change accordingly. For instance, if there are additional required parameters, the interface must allow the user to input and send these values. The pacemaker is also expected to implement an identifier that would allow the DCM to know whether the pacemaker connected is that which is expected. As of assignment 3, support for this functionality is not included in the interface.

The pacemaker at a later data may also send back its parameters to confirm that those configured on the DCM have been correctly sent. The DCM will have to support said functionality.

3 Parameter Sending

3.1 Design Decisions

This module uses a variety of text box objects to take the users input, store the data and send it to the pacemaker after processing it appropriately. The only change to this module since assignment 2 has been the addition of pacing modes and the removal of the egram constant that had been sent alongside the pacemaker parameters.

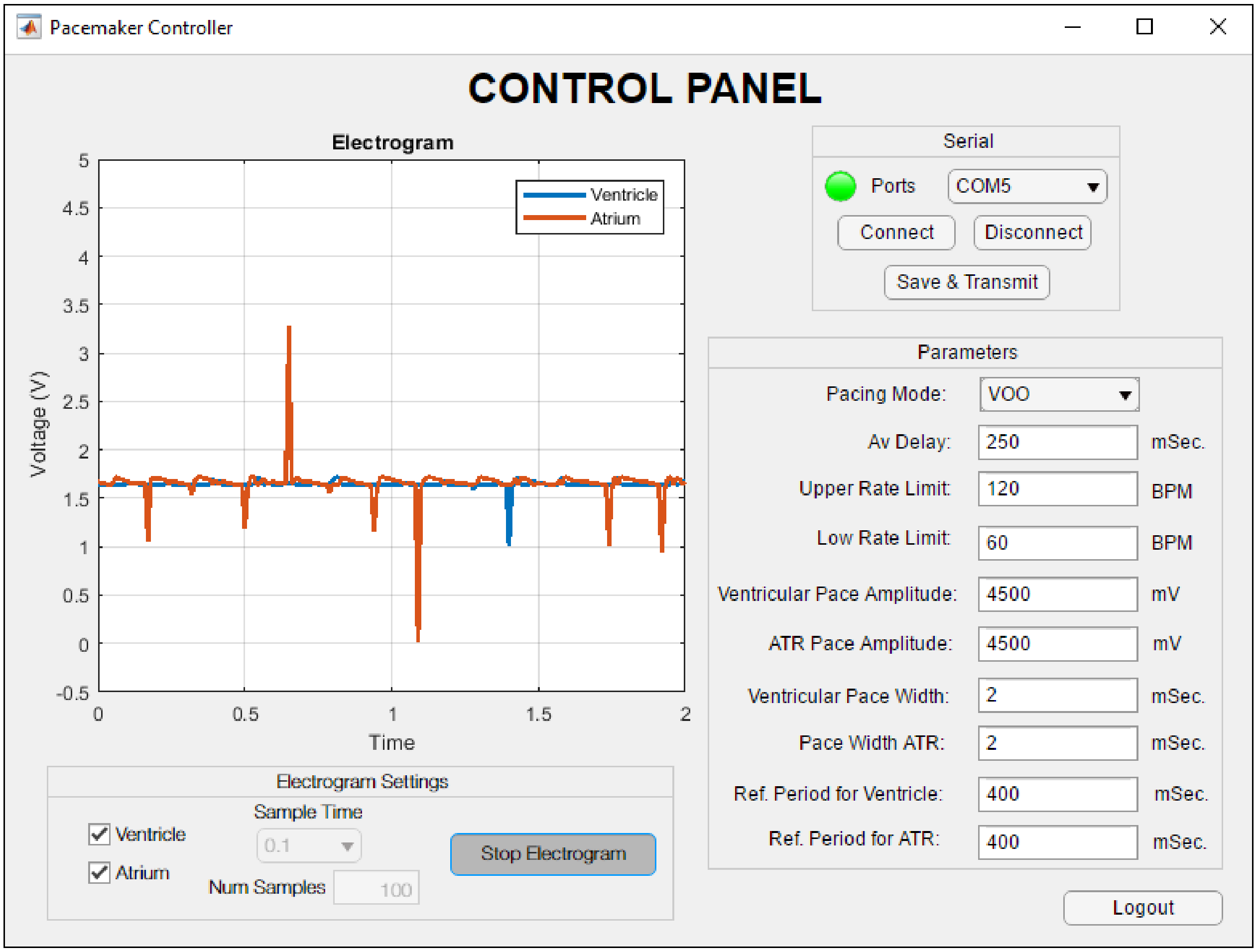
3.2 Future Changes

The manner in which the parameter are sent is subject to change in later designs as presently, much of the users data is trimmed to fit the parameters into an uint8 type. For example, the amplitude of the pace is configured in millivolts and is valid in the range of 500 to 5000. The DCM trims the least significant 2 digits and sends the remainder meaning that this parameter is only accurate to the 100 mV. If greater parameter accuracy is required at a later time, these details can be shifted into additional serial packages and sent to the pacemaker.

4 Electrogram

4.1 Design Decisions

As of assignment 3, the electrogram has been changed to be more intuitive and as such, it has been merged with the main control screen and it is controlled with a simple state button alongside two check boxes which indicate which chambers are to be monitored. A sample egram window is captured below. Clearly the atrium and ventricle are being monitored at the same time and the user can see both chambers beating on the marked graph.



Electrogram Graphing both Atrium and Ventricle Data

Support for electrogram requires multiple global variables that are in the scope of the module that indicate the time step between graphing data and to hold the data to be graphed. A private function is called by a timer every timestep which reads the most recent data from the serial port and graphs it in the electrogram window. This works by shifting data into and out of a array that contains all past egram data for both chambers. Each time the data is graphed, only the chamber(s) which the user has selected are displayed on the electrogram axes.

The serial port buffer is set to be two values in size and upon the call of the timed graphing function, the port data is flushed and a small pause is executed to wait for new serial data. After this pause, the data is pushed into array of graphing data. Next the graph is updated upon realization of the chambers to be monitored. As both atrium and ventricle data is read at all times, both datas are stored regardless of which is to be graphed. Storing both data allows the user to change which chamber is being monitored in real time without stopping the electrogram. Note that the atrium and ventricle is marked on the graph window with a legend.

Both the time step the user wishes to monitor the data at and the number of samples displayed at one time are controllable by the user. Both of these controls are not accessible when electrogram is active thus the electrogram must be stopped prior to changing the parameters.

The black box behaviour of this module is to take the state of the users egram enable button, their graphing parameters, which chamber(s) they want to monitor and the connection status of the serial port as inputs and output in graph form the real time voltage in those monitored chambers.

4.2 Future Changes

As for possible requirements changes, there are a number of details that may be required at a later time. For instance, beat detection could be implemented using hysteresis, where serial data is egram data is compared against previous models such that individual beats may be detected and an approximate observed bpm can be calculated.

Some elements of the graph cannot be considered medically accurate in its present state such as the actual time scale displayed on the graph which is subject to the behaviour of the incoming serial data and the speed of which the data can be updated. As such, even at the highest sample rate of 0.001ms, the DCM struggles to monitor and update the graph at this rate. Additionally, the voltage moniteried is scaled to fit the FRDM-K64F analog to digital conversion and this scaling is reversed on the DCM side however some voltage data is lost and we are unsure whether the ADC is monitoring based off a 5V or 3.3V scale. Greater accuracy of egram may become a requirement at a later stage.