Assignment 2:

1. Part 2
    a. Document how programmable parameters originate at the DCM and are implemented in the device. Show how you can ensure the parameters stored in the Pacemaker are what the doctor input on the DCM. Also justify your choice of the data types used to represent parameters data.

Programmable parameters

Parameter	Nominal	Min	Max	Increment	Unit	Data type	Size (Bytes)
mode	AOO	0	7	1	-	unit8	1
Lower rate limit (lrl)	60	30	50	5	ppm	unit8	1
		50	90	1	ppm		
		90	175	5	ppm		
Upper rate limit (url)	120	50	175	5	ppm	unit8	1
PVARP	250	150	500	10	ms	unit16	2
av_delay	150	70	300	10	ms	unit8	1
reaction_time	30	10	50	10	s	unit16	1
response_factor	8	1	16	1	-	unit8	1
activity_threshold	Med	0	6	V-Low, Low, Med-Low, Med, Med-High, V-High	-	double	8
recovery_time	5	2	16	1	min	unit16	2
MSR	120	50	175	5	ppm	unit8	1
atr_amp	5	0	5	0.1	V	double	8
atr_pulse_width	1	1	30	1	ms	double	8
ARP	250	150	500	10	ms	unit16	2
atr_threshold	0.75	0.25	0.75	-	mV	double	8
		1 mV	10 mV	0.5	mV		
vent_amp	1	0	5	0.1	V	double	8
vent_pulse_width	1	1	30	1	ms	double	8
VRP	320	150	500	10	ms	unit16	2
vent_threshold	2.5	1.0	10	0.5	mV	double	8
		1 mV	10 mV	0.5	mV		
						Total	71

The data type of **uint8** was used for the following parameters: mode, lower rate limit (lrl), upper rate limit (url), av_delay, response_factor, and MSR. This was because the range of the numerical values of said parameters can all be measured within the range of 0 to 255 or 1 byte.

The data type **uint16** was used for the following parameters: PVARP, reaction_time, recovery_time, ARP and VRP. The reason this data type was chosen was because the numerical values of these specific parameters fit into the range of 0 to 65535, or 2 bytes. Both data types (**uint8** and **uint16**) are **unsigned data types**, which means they only consider positive values.

Furthermore, the data type **double** was used for the following parameters: activity_threshold, atr_amp, atr_pulse_width, atr_threshold, vent_amp, vent_pulse_width and vent_threshold. This specific data type was used because it considers the use of decimals in the values of said parameters. **Double** also allows for a larger range of values than a data type “float”.

The parameters stated above are programable and can be set within in the DCM and sent to the board, where they are implemented on the Simulink model with the same data type and size. This is done so that the DCM and hardware can serially communicate with each other, the DCM sending the parameter values configuring the pacemaker.

To check the parameters stored in the pacemaker, we must compare them to the values inputted by the doctor with the DCM. This is done by sending two signals to the pacemaker: One “set” signal, and one “echo” signal. The “set” signal configures the pacemaker, the “echo” requests the Simulink to return the values it is currently configured to. In the DCM we compare the echo signal to the values we expect it to return, IE the values we sent it, verifying that the pacemaker has stored the correct parameters. If there is a mismatch, we throw an error.