

# **Fetal Pacemaker**

## **M. Eng Final Report**

**By  
Manasi Safaya**

**Submitted to:  
Prof. Land**

**May 18, 2006**

## **Executive Summary**

Hydrops fetalis (fetal death) is a life-threatening problem of severe swelling in the fetus that develops when too much fluid leaves the bloodstream and goes into the fetus' tissues. Among several causes that result in hydrops fetalis, fetal arrhythmia is a significant factor. Fetal arrhythmia can be regulated by placing a fetal pacemaker in the mother's womb that will pace the fetal heart at a set rate determined by the physician.

My M.Eng project was to develop a fetal pacemaker whose electronic circuit will be placed in the mother's womb, lead will be placed under the fetus' rib cage and that will provide 2-4 weeks of pacing. The pacemaker will be programmable by an external programmer that will allow the physicians to set the fetal heart rate for each individual fetus.

Due to the complexity of the design, I focused developing the pacemaker circuit and the pacing software. The circuit was developed successfully and met the fetal pacemaker product specifications. The pacing software also successfully paces at a heart rate set by a physician.

The next step in this project (to be continued by another student) is to complete the electronic and mechanical assembly of the pacemaker followed by animal testing to determine if the pulse output from the pacemaker can successfully pace the heart from a set distance (lead under the rib cage).

## **Acknowledgements**

I would like to thank Prof Bruce for his mentorship during the course of this project; without his help I would have been able to complete it. I also would like to thank Dr. Marc Ovardia for introducing me to the concept of developing a fetal pacemaker. Finally, I would like to thank Prof. Lipson for his insight in completing this project.

## **TABLE OF CONTENTS**

	<b>Page</b>
<b>Executive Summary</b>	<b>1</b>
<b>Acknowledgement</b>	<b>1</b>
<b>1. Introduction</b>	<b>3</b>
<b>2. Product Requirements</b>	<b>3</b>
<b>3. Product Specifications</b>	<b>4</b>
<b>4. Detailed Design</b>	
<b>4.1 Pacemaker</b>	<b>5</b>
<b>4.2 Programmer</b>	<b>7</b>
<b>Results</b>	
<b>5.1 Pacemaker Electronic circuit</b>	<b>8</b>
<b>5.2 Software</b>	<b>9</b>
<b>5.3 Output</b>	<b>9</b>
<b>Improvements</b>	<b>10</b>
<b>Future of the Project</b>	<b>11</b>
<b>Appendix A - Product Requirements Document</b>	<b>12</b>
<b>Appendix B – Microcontroller Code</b>	<b>15</b>
<b>Appendix C- Programming procedure for the physician</b>	<b>17</b>

## **1. INTRODUCTION**

Hydrops fetalis (fetal death) is a life-threatening problem of severe edema (swelling) in the fetus and newborn. Hydrops develops when too much fluid leaves the bloodstream and goes into the tissues. There are two causes that result in hydrops fetalis: immune and non-immune. The immune cause is when the mother's immune system recognizes the fetus' red blood cells as foreign and destroys them (Rh incompatibility). Currently, there are prescription drugs available in the market that inhibits the immune response of the mother such that the fetus' red blood cells are not harmed. The non-immune causes however are more prevalent in causing hydrops fetalis with fetal arrhythmia being a significant factor.

Fetal arrhythmia can be present as either Bradycardia in which the fetal heartbeat can fall below 60bpm (beats per minute) or as Tachycardia in which the fetal heartbeat can be greater than 180bpm. Both conditions can eventually result in fetal death (hydrops fetalis) with fetal bradycardia accounting for 5% of all arrhythmias<sup>1</sup>.

Statistics shows that in the USA 1 case per 600-400 and 1 case per 500-1500 internationally develop hydrops fetalis. Furthermore, the mortality rate for fetus with hydrops fetalis is 60-90%<sup>2</sup>.

As fetal arrhythmias can be regulated, the placement of a pacemaker in the mother's womb can help synchronizes the fetal heartbeat thereby reducing the likelihood of it develop hydrops fetalis.

## **2. PRODUCT REQUIREMENTS**

The pacemaker has 2 components, the pacemaker which will pace the fetal heart and the programmer which will be used by physicians to set the heart rate on the pacemaker before it is implanted into the fetus.

The pacemaker will be implanted into the fetus via a 16-18 gauge trocar. Due to the properties of the fetal heart, the pacemaker lead will not be implanted directly on the heart but right under the fetus' ribcage.

After discussing with Dr.Marc Ovadia his vision for the fetal pacemaker and upon literature review on fetal pacing, the product requirements for the pacemaker were determined to be:

- Small size so that it can be injected via a trocar
- Low power (1W output)
- Constant pulse rate that will be set by the doctor through the programmer
- 2-4 week battery life
- Implant technique similar to fetal surgery

---

<sup>1</sup> Wladimiroff JW, Stewart PA, Tonge HM. *Fetal bradyarrhythmia: diagnosis and outcome*. *Prinat Diagn* 1988;8:53-7.

<sup>2</sup> <http://www.emedicine.com/PED/topic1042.htm>

- Implant should not affect the fetus or the placenta

Based on the functionality of the programmer, its product requirements were that:

- Physician should be able to view the heart rate.
- Programmer should ask for conformation before setting heart rate.
- Programmer should attach to pacemaker for programming.

### 3. PRODUCT SPECIFICATIONS

The product specifications for the pacemaker and the programmer were based on the product requirements and components available in the market that met the requirements.

The specifications for the pacemaker are:

<b>Electronic Circuit</b>	
<i>Dimensions:</i>	
Width:	Less than 3.8-4 mm (as the pacemaker will be injected through a trocar)
Length:	Around 5-7mm is acceptable
<i>Design:</i>	
Microcontroller (PIC10F202)	SOT 23 package
<b>Battery</b>	
<i>Dimensions:</i>	
Width:	Less than 3.8 mm
Length:	30-40mm
Voltage:	At least 2V to power microprocessor
Current:	At least 350nA to power microprocessor
Duration:	Should last 2-4 weeks

The specifications for the programmer are:

<b>Electronic Circuit with LCD</b>	
<i>Dimensions:</i>	
Width:	20mm (based on width of LCD screen)
Length:	40mm
<b>Battery</b>	Standard AA battery

For the detailed product specifications, please view appendix A.

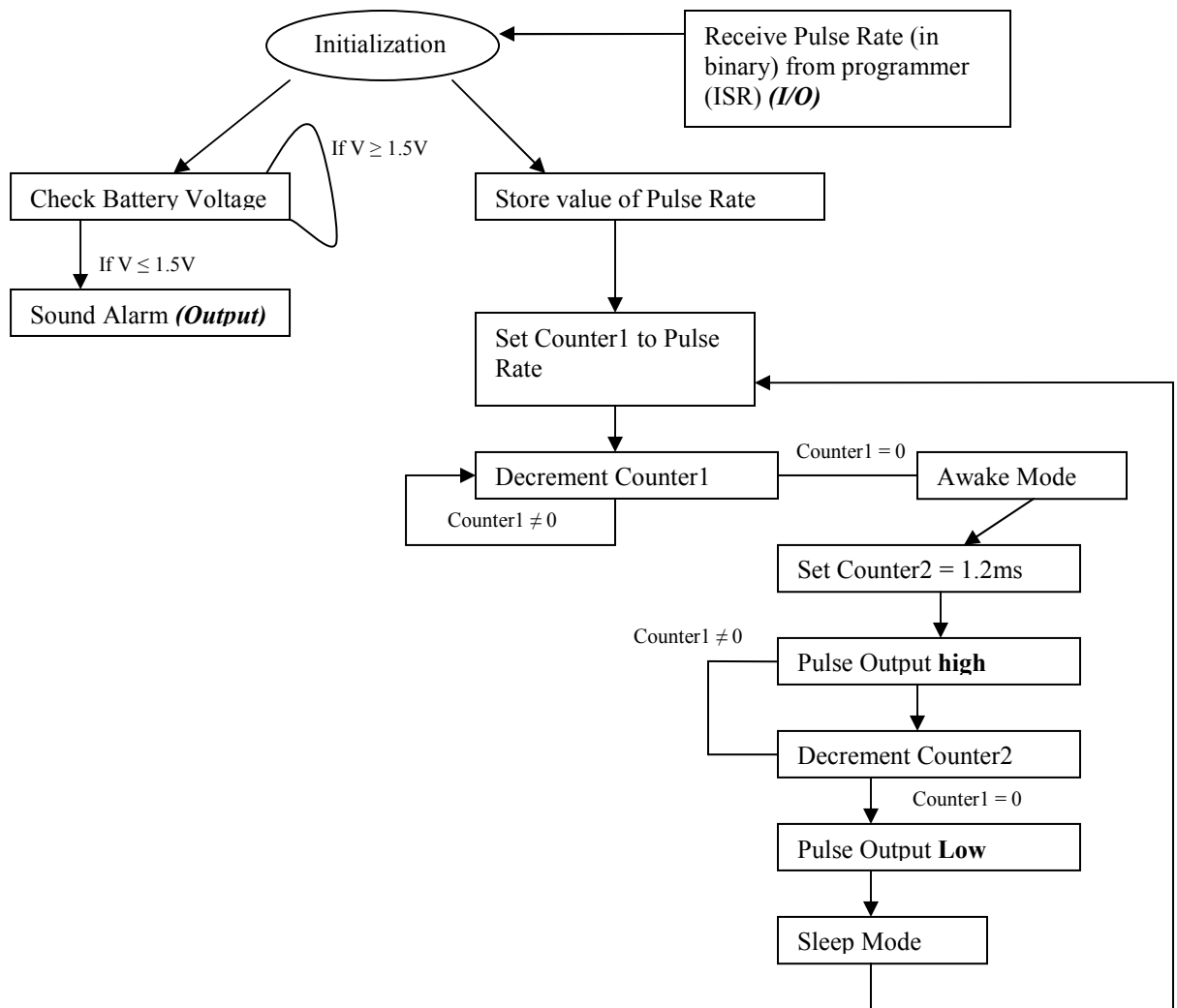
#### **4. DETAILED DESIGN**

The design involves both hardware and software design for the pacemaker and the programmer.

##### **4.1 PACEMAKER**

###### **High Level Software Design:**

The pacemaker software will receive the desired pulse rate (calculated from Formula 1 below) from the programmer and use it to pace the fetal heart. In order to conserve battery life which will be monitored constantly by the circuit, the pacemaker will be in sleep mode between pulses. The pacemaker will continue to pace at the set pulse rate until it detects another value from the programmer. It is to be noted that the pulse rate can only be modified before implantation.



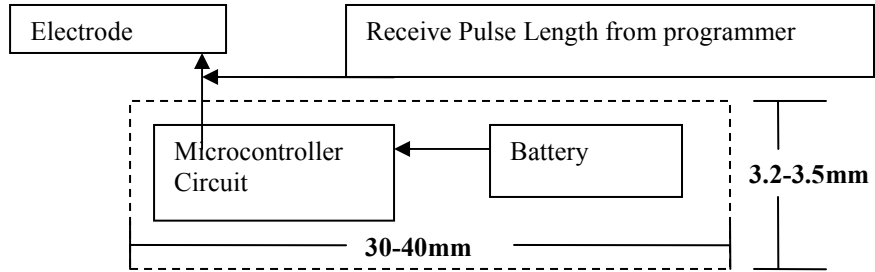
The pulse rate can be calculated using Formula 1:

$$\text{Bpm} = 60000/t \text{ (ms)}$$

The pulse rate of the pacemaker has to be slightly less than  $t$  to ensure that it controls the heart beat.

### High Level Hardware Design:

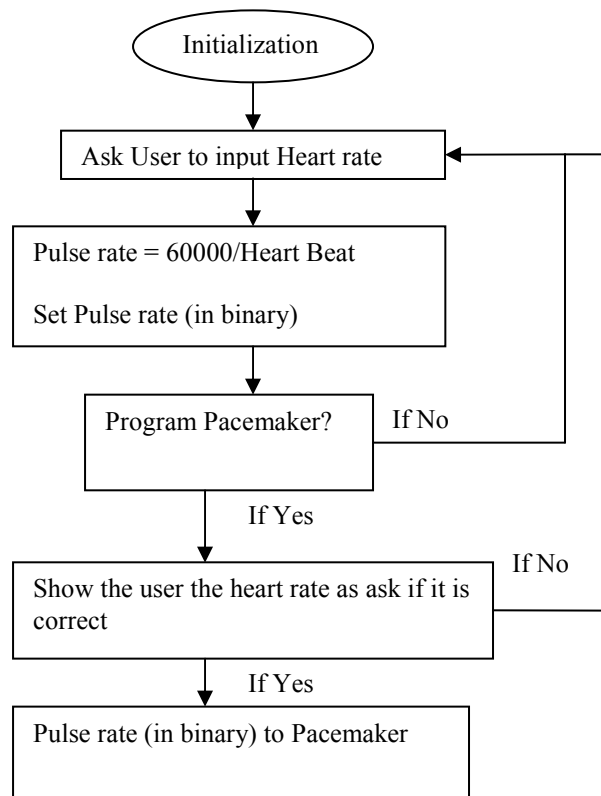
The pacemaker will have one output which will either receive the pulse length from the programmer or send the pulse output to the electrode. The microcontroller and battery will be encased in a titanium alloy casing.



## 4.2 PROGRAMMER

### High Level Software Design:

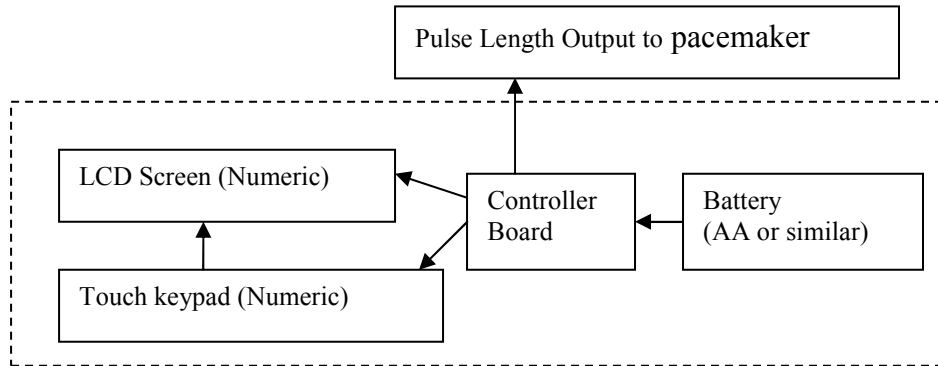
The programmer software will ask the user to input the desired heart rate and calculate the pulse rate which will be converted to a binary value. The user will be first asked to confirm the heart rate and subsequently asked to confirm if he/she would like to output the pulse rate to the pacemaker. The programmer will only be able to set/reset the pulse rate of the pacemaker before implantation.



### High Level Hardware Design:



The programmer hardware will consist of a touch keypad (most likely with arrows) through which the user will be able to set the heart rate. The LCD screen will allow the user to view the heart rate that he/she is setting. The controller will calculate the pulse rate, its binary value and output it to the pacemaker. And standard battery will be sufficient to run the circuit.



## 5. RESULTS

Due to the complexity of the project, I focused my M. Eng project on the electronic circuit and pacing software of the pacemaker.

To develop and test the software, the Microchip Baseline Flash kit was used (Fig 1).

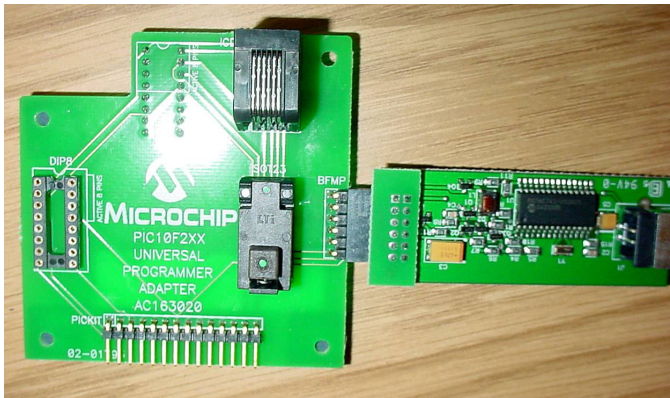


Fig 1: Microchip's Baseline Flash kit for PIC10Fx microcontroller series.

This kit consists of a baseline flash board that received the software code from the computer and the adapter board which is used to program the PIC10F202 microcontroller chip used in the circuit.

### 5.1 PACEMAKER CIRCUIT

The microcontroller is programmable so the physician can set/reset the heartbeat during testing. A capacitor was also placed in the circuit to absorb any transients in the circuit. The circuit layout dimensions complied with the product requirement (Fig 2). The flat

flex header connected circuit to the Baseline Flash kit for programming the microcontroller.

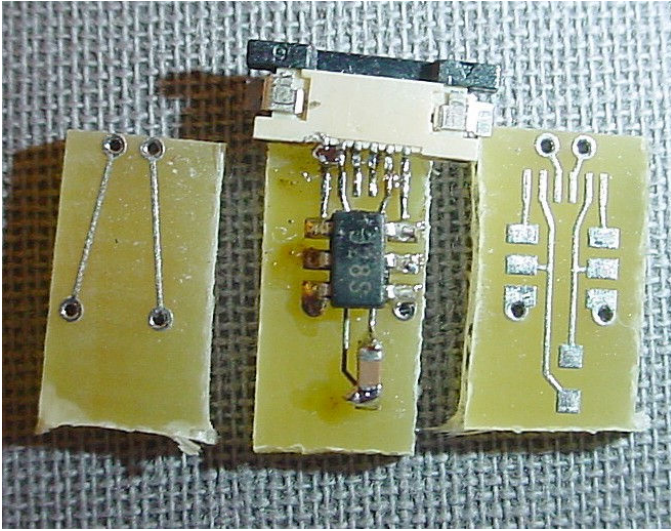


Fig 2: Pacemaker electronic circuit.

## 5.2 SOFTWARE

The software was written in assembly language and has two components to it, the include file (Fetal\_pacemaker\_interface.inc) and the pacing file (Fetal\_Pacemaker.asm).

The include file is an interface file which the physician will use to set the heart rate. Due to the arithmetic abilities of the microcontroller, calculating the rate (formula 1) and converting it into a binary number will be done by the programmer. Therefore, in the include file the physician will have to directly enter the rate as a binary number.

The Fetal\_Pacemaker.asm software is designed to output a 1.2ms pulse “high” pulse and a “low” pulse for the remainder of the pulse width. For example, if the heart rate is set to 120bpm this gives a pulse rate of 500ms. Thus the output will be high for 1.2ms and low for 498.8ms. Also due to time constraints, the pacing circuit does not manage the battery voltage or sleep mode as indicated in the detailed design section. However, once the ideal battery has been chosen for the pacemaker, assembly code for battery management can be added to the software.

Appendix B contains the detailed pacing code and Appendix C contains the programming procedure the physician will use to program the circuit.

## 5.3 OUTPUT

The circuit successfully paces at the heart rate set by the doctor. When the heart rate is set to 120bpm, the circuit pulses every 500 ms (Fig 3).



Fig 3: Pulse Frequency when the heart rate is set to 120bpm

The circuit is also designed to output a 1.2ms pulse with amplitude of 5V (Fig 4).

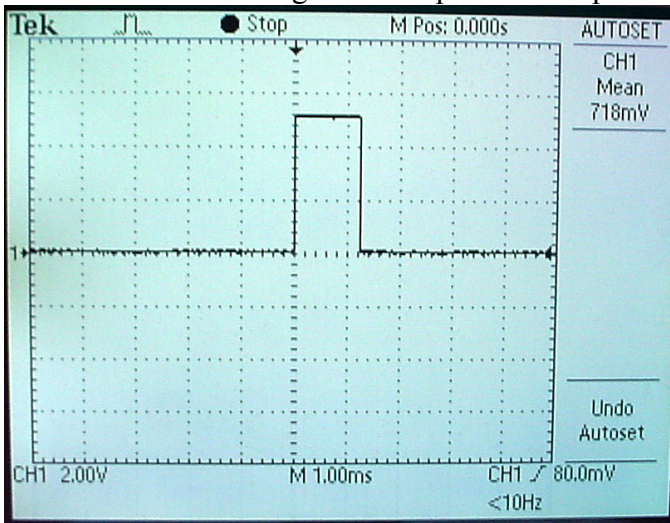


Fig 4: Amplitude of the pulse.

The pulse rate can be changed by varying the heart beat but the pulse width and amplitude of the “high” output is constant.

### IMPROVEMENTS

There are a few improvements that can be applied to the pacemaker. Firstly, in the software the addition of a watchdog timer will put the circuit in sleep mode when the pulse output is low. This will help optimize the battery life and allow it to last longer.

Secondly, a CMOS chip could have been used instead of a microcontroller which may decrease the size of the chip, making it easier to implant through a trocar. Due to the time involved in training and successfully designing a CMOS chip, I opted to use a microcontroller.

Thirdly, the BR425 battery was not tested with the circuit, which could have resulted in some software modifications and addition of code for battery management.

### **FUTURE OF THE PROJECT**

Firstly, after applying the improvements as stated in the previous section, the next step is to complete the electrical and mechanical assembly of the pacemaker. A titanium casing has to be developed for the electronic circuit and battery. Also the ideal lead wire, electrode and connector (to connect the output of the circuit to the lead) have to be identified.

Secondly, the BR425 battery currently being used with the circuit is too large to meet product specifications thus more research has to be conducted to find a smaller battery that meet product specifications.

Last but not the least, once the previous steps have been completed, in vivo (in animals) testing can begin to determine if the pulse rate and amplitude are sufficient to pace the heart from the optimal distance.

## APPENDIX A

### Product Specifications

#### **1. Purpose**

1.1 To define the use and the performance for a Fetal Pacemaker.

#### **2. Introduction**

2.1 The single chamber Fetal Pacemaker is a single use device intended to regulate the heartbeat of fetuses suffering from arrhythmias.

#### **3. Clinical Applications**

##### **3.1. Intended Use**

The single chamber Fetal Pacemaker is a single use device intended to regulate the heartbeat of fetuses suffering from arrhythmias.

##### **3.2. Medical Specialties of users**

Surgeons qualified to perform the electrosurgical procedure, Obstetrician.

##### **3.3. Patient Population**

At risk fetuses 30-35 weeks old.

##### **3.4. Warnings**

Safety of the implantation procedure has not yet been evaluated for safety for the mother or the fetus.

##### **3.5. Accessories or ancillary devices**

Pacemaker programmer which allows the surgeon to set the heart rate before implantation.

#### **4. Medical Rationale**

##### **4.1. Indications**

Intended on assisting as risk fetuses who suffer of bradycardia until delivery.

##### **4.2. Existing Products**

No such product exists in the market

#### **5. Product Description**

##### **5.1. General Description**

The Fetal Pacemaker is a single use device intended to regulate the heartbeat of fetuses suffering from bradycardia.

##### **5.2. Product Requirements**

###### **5.2.1. Microcontroller and Battery Casing**

5.2.1.1. Length of casing should be 40-50 mm.

5.2.1.2. Width of the casing should be 3.8-4.1 mm.

5.2.1.3. Titanium alloy should be used to manufacture casing as it's proven to be biocompatible.

###### **5.2.2. Lead Casing**

5.2.2.1. Length should be 4-5 mm.

5.2.2.2. Width should be 3.5-3.8 mm.

5.2.2.3. A Plastic material should be used to encase lead connection to the microcontroller and battery case.

### 5.2.3. Leads

5.2.3.1. Lead is made from polyurethane and the tip will be a metal alloy.

## 5.3. Battery

5.3.1. Battery should output at least 0.5 mA at 2V.

5.3.2. Battery should have total current of 145mAh to last for 3-5 weeks.

## 6. Engineering Requirements

### 6.1. Physical Requirements

6.1.1. Total length of pacemaker should be 15-20mm in length.

6.1.2. Pacemaker width should be 3.5-3.8mm.

6.1.3. Length of lead is 4-5mm.

### 6.2. Performance requirements

6.2.1. Pacemaker should operate for at least 3-5 weeks.

6.2.2. Pacemaker should be programmable to set the heart rate.

### 6.3. Electrical Requirements

6.3.1. 1W output

## 7. Safety Features

7.1. The product should meet the safety requirements dictated by the FDA and European standards.

7.2. The risks and hazards are:

7.2.1. Fetus:

- Improper implantation technique thereby injuring fetus.
- Electrode not attached at the correct distance from the heart.
- Current from pulse not sufficient to pace fetal heart.
- Leak current may affect other organs.
- Pacemaker entanglement with placenta.

7.2.2 Mother:

- Implantation technique may cause leakage in amniotic fluid.
- Trocar may injure mother and prevent recovery of wound.

## 8. Reliability

8.1. The device should function for at least 3-5 weeks.

## 9. Sterilization Method

9.1. Device will be sterilized at manufacturing plant and must not be removed from packaging until time of implantation.

## 10. Stability and Shelf Life

10.1 The product can be stored anywhere between 0 to 20 degrees centigrade. The shelf life should be at least 2 weeks but will be dictated by the battery life.

## 11. Labeling/Directions for Use

**11.1** Microcontroller and programmer should only be removed from packaging during time of implantation.

**11.2** All labeling will meet requirements for the CE mark.

## **12. Packaging Requirements**

12.1. 12. 1 Both Pacemaker and programmer should be stored in a anti-Static, Sterile bag.

## **13. Regulatory Requirements**

**13.1** Should meet the following requirements:

**13.1.1** ANSI (American National Standards Institute):

- 5840:2005 (Cardiovascular implants) and 60601- 2 (Medical Electrical Equipment).
- DF-21 (Medical Instrumentation)
- MDS-201-0004 (Electromagnetic Compatibility Standard for Medical Devices)

**13.1.2** IEC 60601-1-2:2001 (European standards for Medical Electrical Equipment.

- Part 1: General Requirements for Safety
- Part 2: Collateral Standard: Electromagnetic Compatibility— Requirements and Tests)

## **14. Manufacturing Requirements**

**14.1** Device is still in prototype stage so manufacturing requirements have not been determined.

## APPENDIX B

### INCLUDE FILE

```
;;;;INCLUDE FILE FOR PIC10F MICROCONTROLLER SOFTWARE;;;;;
;;;VERSION 1.0;.....
;;;;WRITTEN BY MANASI SAFAYA, MAY 7, 2006;.....
```

List

nolist

```
.....ONLY MODIFY THIS SECTION IN THE SOFTWARE;.....
```

```
.....SETTING PULSE FREQUENCY;.....
```

```
;;;HOW TO SET Max_count3;...;
```

```
;Set pulse freq: To get pulse rate= (60000/bpm)
; To get Max_count3= pulse rate/2ms
;Convert Max_count3 into binary
```

```
#define Max_count3 B'11111011' ; for heart rate of 120bpm
#define Max_count3 B'10100110' ; for heart rate of 180 bpm
```

list

```
.....END OF CODE
```

### PACING CODE (SIMPLE\_FINAL.asm)

```
;MicroController Program for Fetal Pacemaker
;Developed by Manasi Safaya
;Version 2.1
```

```
.....DO NOT MODIFY BELOW THIS LINE;.....
```

```
list P=10F202
```

```
#include "P10F202.INC"
```

```
#include "PIC10Finclude.inc"
```

```
_CONFIG _CP_OFF & _MCLRE_OFF & _WDT_OFF ; & _IntRC_OSC
```

```
;;;CONSTANT VARIABLES;...;
```

```
#define Max_count1 B'11111011' ;high loop
```

```
#define Max_count2 B'11111011' ;low loop
```

```
.....SETTING COUNTER VARIABLE;.....
```

```
cblock 0x08
```

```
counter1
```

```
counter2
```

```
counter3
```



```

endc

,,,,,SETTING START OF PROGRAM,,,,,
;Reset
    ORG 0000h                ; processor reset vector
    nop
    ; required by in circuit debugger
    goto Mainline           ; go to beginning of program

,,,,,,,,,,,,,,,,,MAIN PROGRAM,,,,,,,,,,,,,,,,,

Mainline
    movlw B'1000'           ;1=input 0=output GP2 is T0CK1
    TRIS GPIO               ;Sets GPIO as 1101 so set GP1 as output.
    movlw Max_count1
    movwf counter1
    movlw Max_count2
    movwf counter2
    movlw Max_count3
    movwf counter3
    call High_Output
    call Low_Output
    goto Mainline

,,,,,SETTING HIGH OUTPUT,,,,,,,,,
; This is constant with a pulse duration of 1.25ms

High_Output
    movlw 15                ;High Output
    movwf GPIO
    decfsz counter1,1       ;Decrement and skip if 0
    goto High_Output
    return

,,,,,SETTING LOW OUTPUT,,,,,,,,,

Low_Output
    movlw 0
    movwf GPIO
Low_loop1:    nop           ;2ms inner loop
              nop
              nop
              nop
              decfsz counter2,1 ;Decrement and skip if 0
              goto Low_loop1
    decfsz counter3,1
    goto Low_Output
    return

end

```

## APPENDIX C

### PROGRAMMING PROCEDURE FOR THE PHYSICIAN

#### **Initial Steps:**

1. The microcontroller code will be given to Dr.Ovadia on a Cd. Copy the folder from the cd onto a directory (any you prefer) on the computer.
2. Download and install the microchip IDE software on the CD, *MPLAB IDE v7.30 Full Zipped Installation*. Follow the instructions in the software to install properly.
3. Install programming software from the *MPLAB® Integrated Development Environment* CD-ROM

<b>Step</b>	<b>Procedure</b>
1	Once the MPLAB IDE v7.30 software has been installed. In the IDE go to Project → Open Then open PIC10F.mcp from the location where it its stored.
2	Once the project is opened to go the PIC10F.mcw window where you can see the source files and include files.
3	Click on the include file Fetal_pacemaker_interface.inc to open.
4	In the include file go to the section “Setting pulse frequency”. To set the heart rate, the variable Max_count3 has to be set. To do so, use the formula: $Max\_count3 = (60000/(2*heart\ rate))$
5	Once the Max_count3 value has been calculated, convert that into binary. This can be done by using the calculator on a PC. In Microsoft windows, go to Start→Programs→Accessories→Calculator. Then go to View →Scientific, and enter the value for Max_count3 and then click on “Bin”. This will give you the binary value for Max_count3
6	Set the binary value of Max_count 3 in the include file.
7	Save the file. To compile go to Project →Build All
8	Open Pick1 flash starter kit (which you installed from the CD) In the window, for the “Device” select PIC10F202
9	Go to File →Import hex Go to the folder where the micro controller code is stored and import Fetal_Pacemaker.hex
10	Connect the hardware as shown in Fig 1. The cable should be connected to the USB port on the computer and the other end to the USB connector on the Baseline flash board.
11	In the Pick1 flash starter kit window, click on write to device.
12	The microcontroller chip should now be pacing the heart rate that was set.