KAP MEDICAL INC.

Functional Electronics Description: Autofowler

Release Date: 5/03/2002

Overall Description:

The Autofowler function allows the mattress to wirelessly communicate a change in position to its controlling unit. When a patient is laying on the bed in a horizontal position, the entire body is fully supported by the therapeutic mattress. When the head/torso portion of the bed is tilted at an angle, usually above 45 degrees, the distribution of weight will shift and center around the lower torso/hip are of the patient (This occurs when the patient is articulated at an angle to eat a meal or watch T.V. etc). Since the weight is now concentrated in a smaller area in the center of the bed, the patient may have the tendency to egress further into the mattress where therapy is no longer effective. The caregiver can compensate for this by increasing the overall pressure in the mattress which will raise the patient back to a position where therapy is active again. There are many instances, however, when a caregiver may not be readily available to perform this function. In order to insure that active therapy is always applied when possible, a low power wireless transmitter will be placed inside of the mattress. This transmitter assembly will detect when the head of the bed is at an angle great enough to cause unwanted egress, and it will transmit this state to the controlling unit.

The transmitter will use a microprocessor and a QwickRadio transmitter chip manufactured by Micrel to assess and send this data to the control unit. The control unit will use the Micrel QwikRadio receiver chip to receive the information and perform any necessary adjustment to the air pressure in the mattress.

Function of the Transmitter Unit:

1) The Transmitter will transmit at the 315MHz frequency which falls within the ISM band.
2) The output data rate will be 1KHz or less (currently set at 868Hz)
3) The transmission will take place at random intervals only when a change in fowler position is detected
4) The transmission will be active for 4 seconds or less when transmitting status data
5) When not transmitting, the transmitter IC will be in a SHUTDOWN/LOW POWER mode.
6) The electronics will wake up every 4 seconds and check for a change in the detected angle
    a. The last transmitted status was fowler position: unit looks to see if angle has changed and is now less than the fowler angle:
        i. If less than the fowler angle, the unit will enable the transmitter , and then transmit the low angle status to the receiver
        ii. If still higher than the fowler angle the unit does not transmit any information
        iii. Software executes sleep commands as described in "7)"
    b. The last transmitted status was bed in flat (horizontal) position: unit looks to see if angle has changed and is now greater than the fowler angle:
        i. If now greater than the fowler angle, the unit will enable the transmitter, and then transmit the greater angle status to the receiver
        ii. If still lower than the fowler angle the unit does not transmit any information
        iii. Software executes sleep commands as described in "7)"
7) The microprocessor makes sure that the transmitter is off, resets the 4 second wakeup time and then puts all of the electronics in to sleep mode

Function of the Receiver PCB:

1) The receiver PCB will continually scan for transmitted data which will include the serial number of the transmitter. Only one unique transmitter serial number will be accepted by each receiver.
2) If the serial number does not match the number stored in the receiver the unit ignores the transmitted data
3) If the serial number does match the stored number in the receive:
    a. Fowler *Active* status transmitted (head of bed above fowler angle): Unit will increase the air pressure in the mattress by a pressure to be determined by the control unit. If this is a turning unit, the turn function will be shut off and disabled.
    b. Fowler *Inactive* status transmitted (head of bed below fowler angle): Control unit will decrease the pressure in the mattress by a pressure to be determined by the control unit. If this is a turning unit, the turn function will be enabled if the user wishes to enter that mode of operation.

Circuit Boards:

The Autofowler function is accomplished with two circuit boards. They are both standard double sided, FR4, .062" copper clad PCBs. The two PCBs are: **Transmitte**r PCB and a matched **Receiver** PCB. Both PCBs will be populated with surface mount components (SMT) but may also have through hole components mounted on the PCBs such as connectors.

Transmitter PCB

1) Transmits fowler data using the MICREL MICRF112 QwikRadio transmitter IC using the ASK (Amplitude Shift Keying) protocol
2) The antenna used for transmission will be etched in to the PCB as per MICREL specifications
3) The microprocessor will be connected to a mercury switch or interfaced to a MEMS accelerometer to determine if the transmitter has been tilted above or below the fowler angle
    a. If a Mercury Switch is used, the switch angle will be set during manufacturing to enable the mercury in the switch to activate or de-activate when the desired angle is reached
    b. If a MEMS sensor is utilized, the unit may also include a 2 position DIP switch which will be accessible to the user. These switches will enable the user to externally select between 4 different tilt angles in which the fowler angle will be detected
4) A 3.6V Lithium-Ion battery will solder to the PCB and provide the power to run the microprocessor and the transmitter IC
5) The battery will be capable of operating the transmitter PCB for 1 to 2 years of normal operation
6) After assembly and test, the transmitter PCB will be placed in a potting shell and the contents will be potted using an electrically isolated potting compound (such as Insulcast INS116FRFCFS)

Receiver PCB

1) Receives data using the MICREL MICRF010 QwikRadio receiver IC
2) Determines the following criteria:
    a. Is the data from the transmitter which matches it's internal serial number
    b. Is the receiver being matched to a new transmitter (learning a new serial number)
    c. Is the unit above the fowler angle
    d. Is the unit below the fowler angle

3) The antenna for the receiver can be either an external antenna attached to the PCB, or the antenna can be etched in to the PCB in a HELIX format
4) There are two data lines from the microprocessor which will interface with the host main electronics
    a. OUTPUT:
        i. Logic 0 = Fowler Mode Active
        ii. Logic 1 =Fowler Mode Inactive
    b. LEARN:
        i. Logic 1 = Learn mode Inactive, receiver operates normally
        ii. Logic 0 = Unit waits for new transmission from Transmitter and records the received serial number from the transmitter . This is now the only transmitter data that it will respond to.
5) The receiver will get it's power from the host system through a 4-Pin connector
    a. Pin1: POWER (+5v to +5.4V)
    b. Pin2: LEARN
    c. PIN3: OUTPUT
    d. PIN4: Ground

Microcontrollers:

The transmitter PCB will use the PIC18LF2221-I/SO microcontroller, manufactured by Microchip. The controller will enable or disable the transmitter and encode any data to be transmitted and send that data to the MICRF112 transmitter IC. This controller has a built in I2C interface module which will be used to interface to a MEMS device if that is the method selected to determine the tilt angle. If a 2-position DIP switch is used, the controller will look at the switch position when it wakes up and then determine which fowler angle threshold it will be looking for. This controller will operate at +5v but will also be able to operate down to +2.5V. This will allow the transmitter to fully utilize all of the stored energy in the Li-Ion battery.

The Receiver PCB will use the PIC12F629-I/SN microcontroller, manufactured by Microchip. This IC will receive the data from the MICRF010 chip, decodes the data, and determines any necessary action to perform utilizing this data. The IC will also look at the control signal from the host controller to determine if it needs to learn a new serial number from the transmitter it will be matched to.

Firmware Description:

Transmitter PCB

The firmware for the PIC18LF2221-I/SO will be written in Assembly Code per the Microchip RISC specifications and instruction set. The PCB will have a 6-pin connector that will allow for ICSP (In-Circuit-Serial-programming). The microcontroller will be soldered on to the PCB and programmed through this socket after the PCB is assembled.

The Firmware source code will be saved under FWL_XMT_MEMS.ASM

The output file generated for programming the PIC18LF2221-I/SO will be FWL_XMT_MEMS.HEX

The internal watchdog timer will be enabled allowing the transmitter to reset itself if the software stops execution for any reason.

The unique serial number will be generated for each transmitter by using a SQTP file within the MPLAB IDE programming environment. The unique, 3 byte, identifying serial number will be stored in the program at memory locations 0x800, 0x801 and 0x802. This data will be loaded in to the transmission buffers before a transmission to the receiver.

The Configuration bits which set up the initial operation of the microcontroller will be defined in the beginning of the software program as follows:

```
        LIST  P = 18F2221
;
    CONFIG      OSC=INTIO2, FCMEN=ON, IESO=OFF, PWRT=ON, BOR=OFF, WDT=ON, WDTPS=1024, MCLRE=ON
    CONFIG      LPT1OSC=OFF, PBADEN = DIG, STVREN=ON, LVP=OFF, BBSIZ = BB256, XINST=OFF
    CONFIG      DEBUG=OFF, CP0=ON, CP1=ON, CPB=OFF, CPD=OFF, WRT0=OFF, WRT1=OFF
    CONFIG      WRTB=OFF, WRTC=OFF, WRTD=OFF, EBTR0=OFF, EBTR1=OFF, EBTRB=OFF
```

On power up (Li-Ion battery is initially soldered to the PCB) the program will branch to "INTLZ" and perform the following functions:

1) Setup I/O ports A,B,C for all individual pins as either an input or an output
2) Activate internal pull-up resistors on all PORTB inputs
3) Set internal oscillator for 4MHz
4) Setup A/D pins as all digital (no A/D inputs)
5) Set up internal timers as follows:
    a. TIMER0 – Generates interrupts for Interrupt routine execution
    b. TIMER1 – General purpose timer used within the main program and subroutines
6) Initialize I2C MSSP module for communication with MEMS Accelerometer
7) Program jumps to self test before executing the main program. Microcontroller attempts to communicate with the MEMS IC through the I2C port:
    a. If communication is not verified: The program loops back to the communication test and will re-try communication attempt. The program will not continue until communication is successful.
    b. If communication is verified: The Green LED on the PCB will flash for 10 seconds, indicating proper communication between the microcontroller and the MEMS IC.

After power up initialization and I2C test, the firmware will jump begin execution of the main program as follows:

1) Power up the MEMS accelerometer and get current angle measurement
    a. Check to see if calibration is being requested (calibration request PCB pins are shorted). If yes then the program will jump to step 5.

b. If no then check to see if a valid angle found
    i. If no then go to step 4. Put unit to sleep and check again in 4 seconds.
    ii. If yes, then look at DIP switch inputs to determine which target fowler angle to look for
        1. If DIP switch is not implemented then the unit will default to 45 Degrees

2) Compare the current angle with the target angle. Determine if the fowler angle has changed (up or down) enough to cause a change in the fowler status (active or inactive)
    a. If no, then go to step 4 and put unit to sleep
    b. If yes then continue status transmission
3) Transmit data to receiver:
    a. Load SBIT register with a data byte signifying this transmission is from an autofowler transmitter (0x0A3)
    b. Load serial number registers with the serial number of this transmitter
    c. Send pre-amble to wake up receiver and indicate new data is being sent
    d. Send first byte signifying this transmission is from an autofowler transmitter (0x0A3)
    e. Transmit unique serial number register data
    f. Transmit "Fowler Active" or "Fowler Inactive" data bit
    g. Repeat steps "d through f" until 47 transmission have been completed
4) Make sure that the transmitter is off, resets the 4 second wakeup time and then puts all of the electronics in to sleep mode

5) CALIBRATION
    a. <u>INITIATE CALIBRATION FOR DUT (DEVICE UNDER TEST):</u>
        i. Place PCB on 15 Degree angle ledge of KAP tool #KM-088. Using a pair of tweezers, short the two pins just above the battery marked "QC TEST"
            1. The GREEN LED will flash **1 time** to indicate the 15 Deg. angle has been captured.
        ii. Place PCB on 25 Degree angle ledge of KAP tool #KM-088. Using a pair of tweezers, short the two pins just above the battery marked "QC TEST"
            1. The GREEN LED will flash **2 times** to indicate the 25 Deg. angle has been captured.
        iii. Place PCB on 35 Degree angle ledge of KAP tool #KM-088. Using a pair of tweezers, short the two pins just above the battery marked "QC TEST"
            1. The GREEN LED will flash **3 times** to indicate the 35 Deg. angle has been captured.
        iv. Place PCB on 45 Degree angle ledge of KAP tool #KM-088. Using a pair of tweezers, short the two pins just above the battery marked "QC TEST"
            1. The GREEN LED will flash **4 times** to indicate the 45 Deg. angle has been captured.
        v. Return DUT to the flat (non-fowler) position on the testing table.

    b. <u>TEST CALIBRATED VALUES:</u>
        i. Tilt DUT until it rests on the 15 Degree Ledge. The Green LED will light to indicate the 15 Degree angle was detected. The LED will stay lit.
            1. Return DUT to the flat (non-fowler) position on the testing table. The LED will turn off when the angle becomes less than 10 Degrees.
        ii. Tilt DUT until it rests on the 25 Degree Ledge. The Green LED will light to indicate the 25 Degree angle was detected. The LED will stay lit.
            1. Return DUT to the flat (non-fowler) position on the testing table. The LED will turn off when the angle becomes less than 10 Degrees.

iii. Tilt DUT until it rests on the 35 Degree Ledge. The Green LED will light to indicate the 35 Degree angle was detected. The LED will stay lit.

1. Return DUT to the flat (non-fowler) position on the testing table. The LED will turn off when the angle becomes less than 10 Degrees.

iv. Tilt DUT until it rests on the 45 Degree Ledge. The Green LED will light to indicate the 45 Degree angle was detected. The LED will stay lit.

1. Return DUT to the flat (non-fowler) position on the testing table. The LED will turn off when the angle becomes less than 10 Degrees.

c. TEST DUT TRANSMITTION:

i. Tilt DUT until it rests on the 45 Degree Ledge. The Green LED will light to indicate the 45 Degree angle was detected. The LED will stay lit. The unit will transmit it's ***Fowler Active*** code for 4 seconds, then the LED will turn off.

ii. Return DUT to the flat (non-fowler) position on the testing table. The Green LED will light to indicate the non-fowler angle was detected. The LED will stay lit. The unit will transmit it's ***Fowler Inactive*** code for 4 seconds, then the LED will turn off.

d. CALIBRATION/TESTING COMPLED:

i. The unit will automatically go back into the sleep state after the fowler inactive code is transmitted and the LED turns off (step c.ii). Unit is now ready for potting and normal operation.

NOTE: If the unit is not tilted Up/Down within any 1 minute period, the test program will time-out. The test/calibration program will be aborted, and the unit will return to sleep mode.

Transmitter Subroutine Definitions:

1) **GET_X** [Communicates with MEMS accelerometer and retrieves angle data for X-Axis. Returns the average of 10 data readings]
    a. **AVERAG** [Calculates the average of the 10 data readings)
2) **RESTOR** [Restores all saved registers form non-volatile memory to their active working registers]
3) **HLFSEC** [Delay loop of .5 seconds]
4) **TGLLED** [toggles green LED during power up and calibration]
    a. **FDELAY** [calculates delay between LED toggle]
5) **DLY30** [Executes a 30mS delay]
6) **DLY10** [Executes a 10mS delay]]
7) **DELAY** [Executes a 385uS delay]
8) **SNDI2C** [Sends data stored in the SSPBUF register via the I2C port]
9) **I2CSTR** [Sends an I2C start bit via the I2C port]
10) **I2CRTR** [Sends an I2C re-start bit via the I2C port]
11) **I2CSTP** [Sends an I2C stop bit via the I2C port]
12) **RECI2C** [Receives data via the I2C port and loads data in to the SSPBUF register]
13) **WATI2C** [delays until the I2C data has been sent or received and then clears the data completed flag]
14) **MEMWRT** [Writes data/control information to the MEMS accelerometer]
15) **MEM_RD** [Reads data/control information from the MEMS accelerometer]

16) **POUT** [Transfers the 8-bits of data stored in the DATOUT register to the RF transmitter IC]
17) **RSTTMO** [Resets the timeout count used to exit the calibration routine if no actions are performed within a 1 minute time frame]
18) **NO_KEY** [Debounces the PCB pad contacts used in the QC/Calibration routine]

Receiver PCB

The firmware for the PIC12F629-I/SN will be written in Assembly Code per the Microchip RISC specifications and instruction set. The PCB will have a 4-pin connector and one extra PCB pad that will allow for ICSP (In-Circuit-Serial-programming). The microcontroller will be soldered on to the PCB and programmed through this socket after the PCB is assembled. The receiver PCB will get its +5V power from the host PCB through this 4-pin connector. The connector will also be used to supply the data interface between the host microcontroller and the receiver PCB.

The Configuration bits which set up the initial operation of the microcontroller will be defined in the beginning of the software program as follows:

```
LIST P = 12F629

__CONFIG 0x014C
```

On power up the program will branch to "INTLZ" and perform the following functions:

1) Setup I/O port using the TRISIO register. This will set all individual pins as either an input or an output.
2) Activate internal pull-up resistor on output pin GPIO1
3) Dectivate internal pull-up resistor on output pin GPIO5
4) Set internal oscillator for 4MHz
5) Initialize all working registers to a know state
6) Default fowler output on pin GPIO1 as fowler inactive (Logic 1)
7) Delay .5 seconds to allow power to settle
8) Restore saved transmitter serial number from non-volatile memory

After power up initialization the firmware will begin execution of the main program as follows:

1) Check for learn mode active by checking pin GPIO,1
    a. If active, reset Fowler output to host as fowler inactive (GPIO,0 = 1)
2) Filter random RF noise by looking for at least 20mS where the input is low (GPIO,1 = 0). If not low for a continuous 20mS, then return to step 1
3) Look for start bit (GPIO,5 = 1) within 20mS. If not high within 20mS the return to step1
4) Receive Transmitter serial number and data
    a. If all data was not received correctly return to step 1
5) Check to see if transmitter serial number matches this receiver
    a. If yes: go to step 6
    b. If no:
        i. If Learn Mode is active go to step 6
        ii. If Learn Mode is not active return to step 1
6) Adjust output on pin GPIO,0 to reflect fowler/no fowler data from receiver
        i. If Learn Mode is active go to step 7
        ii. If Learn Mode is not active return to step 1

7) Move the received transmitter serial number to Receiver serial number registers (RNUM1, RNUM2, RNUM3)
8) Store new serial number in non-volatile memory
9) Return to step 1

Receiver Subroutine Definitions:

1) **DELAY** [Executes a 585uS delay]
2) **LOOKHI** [Looks for input data transition from Low to High on pin GPIO,5. If no Low to High transition is detected within 585uS, the routine will return to the main program]
3) **LOOKLW** [Looks for input data transition from High to Low on pin GPIO,5. If no High to Low transition is detected within 585uS, the routine will return to the main program]

____

Dan Rosenmayer / Senior Design Engineer
KAP Medical LLC.