3D PaintBrush

Executive Summary

We will be implementing a paint brush through the use of embedded systems. This will recognize and transmit location information alongside the angle to render strokes on a GUI. These strokes will recognize motions made in a 2D plane and update the GUI which will be able to be shown on any web browser capable digital display. The idea here is to make this as usable and comfortable to use as a tablet for art, but without any physical device actually being drawn on. This will be accomplished through a paint brush peripheral alongside and FPGA used in tandem to record the physical location and the changes of it in order to recreate the motions made in an effort to draw, digitally.

Project Description

This paint brush will have a gyroscope and an accelerometer that will be used in combination to calculate an accurate angular position. This process is called sensor fusion and is used in many applications to retrieve an angular position. The gyroscope itself outputs an angular velocity. The gyroscope data integrated over time will provide an estimate of current angular position. The data from the gyroscope is prone to drift over time because of the imperfections the device has due of the manufacturing process. This drift offsets the output values of the devices in relation to the severity of the imperfections. Another inaccuracy can rise from quick moments that may not get factored into the integration because they fall between the polling rate, or granularity, of our integration. These factors will result in a drift in our calculated value of the angular position compared to the actual angular position. This drift will be combated and corrected by the accelerometer, which always has an acceleration vector reference that points to the center of the earth due to gravitational forces. Based on the current integration's summed value from the gyroscope, which is a ballpark idea of what the angular position is, a correction factor will be determined from the relationship between the current summed value and the ground acceleration vector and added to this summed value. Using the accelerometers reference to ground in combination with trigonometry to obtain the angular position is unusable due to the sporadic nature of accelerometers. The data retrieved from accelerometers is prone to large unwanted spikes due to the nature of the electromagnetic device, which all will be combated by the use of sensor fusion. The FPGA will receive the the values from the accelerometer and gyroscope and will execute the mathematics involved in the integration and correction mechanics. The running sum of the angular position will reside in a register on the FPGA that can be accessed through software.

This paintbrush will also include ultrasound locating features. This will be accomplished with two ultrasound receivers that will be placed in front of the embedded system paint brush, where the brush will be equipped with an ultrasound transmitter. A state machine will start on the FPGA that transmits an ultrasound wave to the receivers from the brush, while also starting timers on the FPGA that are waiting to be stopped by a signal received response from the ultrasound receivers. When this state machine ends these timer values go through a series of mathematical manipulation in the FPGA that use the speed of sound and how long it took for the transmitted wave to reach each receiver to obtain the distance the transmitter is from each brush and then using these distance values to 'triangulate' the position of the transmitter onto a mapping of a 2D cartesian plane. The current calculated X, Y position of the transmitter will reside in registers on the FPGA that can be accessed through software.

The aforementioned technologies used to determine location will ultimately end up with the software running on the Altera board receiving X, Y, angular position as well any buttons currently being pressed. With this information we will connect to a remote server via http post and get requests to properly track movements and inputs and keep a queue of them. The server itself will have Javascript that will render a GUI on a web browser connected to the server. This will alleviate the process of rendering the graphics from our board but instead borrow the resources of whatever device is currently being used to send a signal to the digital display. The GUI itself will have different modules to show the queue of recently done actions, which will be able to be undone, as well as displays to show the selection of either thickness or color. The server will run at 60 frames per second, as that is the native refresh rate on the vast majority of panels produced, as well as not be too strenuous on our board itself.

Project Features

- 4 Buttons plus LEDs
    - Stroke button
        - When pressed will toggle whether a stroke is being recorded
    - Color Button
        - Rotating the pen clockwise or counter clockwise will go through a color wheel
        - The color wheel will be represented with an RGB Led
    - Thickness Button
        - Similar to the RGB Led we will have 5 LEDs which will serve as a scale of how thick the stroke will be
    - Undo Button
        - Pressing it will undo the most recent option, whether it be color, stroke or thickness
    - PCB integration will be done by Matt by Week 8 (2 hours)
    - FPGA integration will be done by Mike by Week 10 (2 hour)
    - Software integration (packets) will be done by Marshall by Week 11 (10 hours)

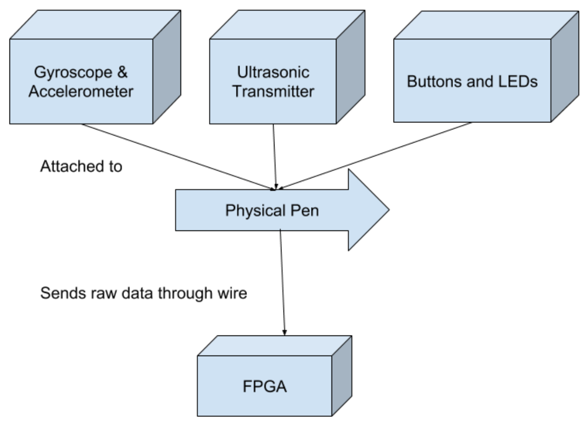
- I2C Communication between pen and FPGA
    - FPGA will be able to poll buttons on pen through I2C bus
    - FPGA can update LEDs on pen through I2C bus
    - PCB integration will be done by Matt by Week 8 (6 hours)
    - FPGA integration will be done by Mike by Week 10(2 hours)
- Ultrasonic receivers
    - Will provide a circumference in which location will be accurately traced
    - Will require configuration pending positioning chosen so FPGA calculations are consistent and can be used in different settings
    - PCB integration will be done by Matt by Week 8 (6 hours)
    - FPGA integration will be done by Mike by Week 10(20 hours)
- Gyroscope and Accelerometer
    - Will improve overall accuracy of location as well as provide angle information for the aforementioned selection wheels.
    - PCB integration will be done by Matt by Week 8 (2 hours)
    - FPGA integration will be done by Mike by Week 10(10 hours)
- Web Server
    - Real time updates from microprocessor to server with HTTP
    - Server provides update to website session
    - VPS integrated to project by Matt by Week 12 (3 hours)
- Ease of display
    - The FPGA will send data packets to a remote server which will be able to be brought up by a web browser.
    - This will allow a range of displays to be used, provided they can access and internet browser.
    - Drawing motions
    - Context menus for switching thickness and color
    - Will be coded by Marshall with Matt into FPGA by Week 11 (20 hours)

Cost Estimate

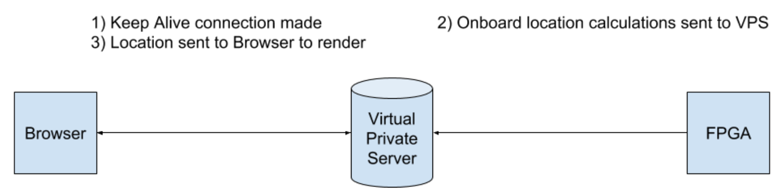
Piece of equipment	Total Price
Power Board	$20
Pen, Ultrasound Receiver, & Phone Cable PCB	$30
Gyroscope + Accelerometer	$5
2 X Ultrasonic Receiver	$3
Ultrasonic Transmitter	$1.5
Passive Components	$30
Pen Microcontroller	$2
Phone Cable	$2
Total	$93.50

Block Diagrams

Physical Paintbrush



Client - Server



Ultrasonic Receiver