RF-based Long-range Motion Recognition and Communication System

Spring 2023

Joe Luo (luo42)

Zekai Zhang (zekaiz2)

James Tian (zeyut2)

I. **Introduction**
   1. Problem
   2. Solution
   3. Visual Aid
   4. High-level requirements list
II. **Design**
   1. Block diagram
   2. Subsystem Overview
   3. Subsystem Requirements
   4. Tolerance Analysis
III. **Ethics and Safety**
IV. **References**

Introduction

Problem

While society accelerates into the digital age, an avid demand for more aggressively intimate ways of communication rises especially as the world welcomes a post-covid traumatic recovery. We witness the emergence of many novel products, albeit with mixed reception, that embrace this new concept, such as VR games, Metaverse, and holographic projection. It's apparent that in modern days we crave information that goes beyond texts, videos, and sounds, but something mobile, three-dimensional, and interactive – for instance, transferring and reproducing motion across a long distance.

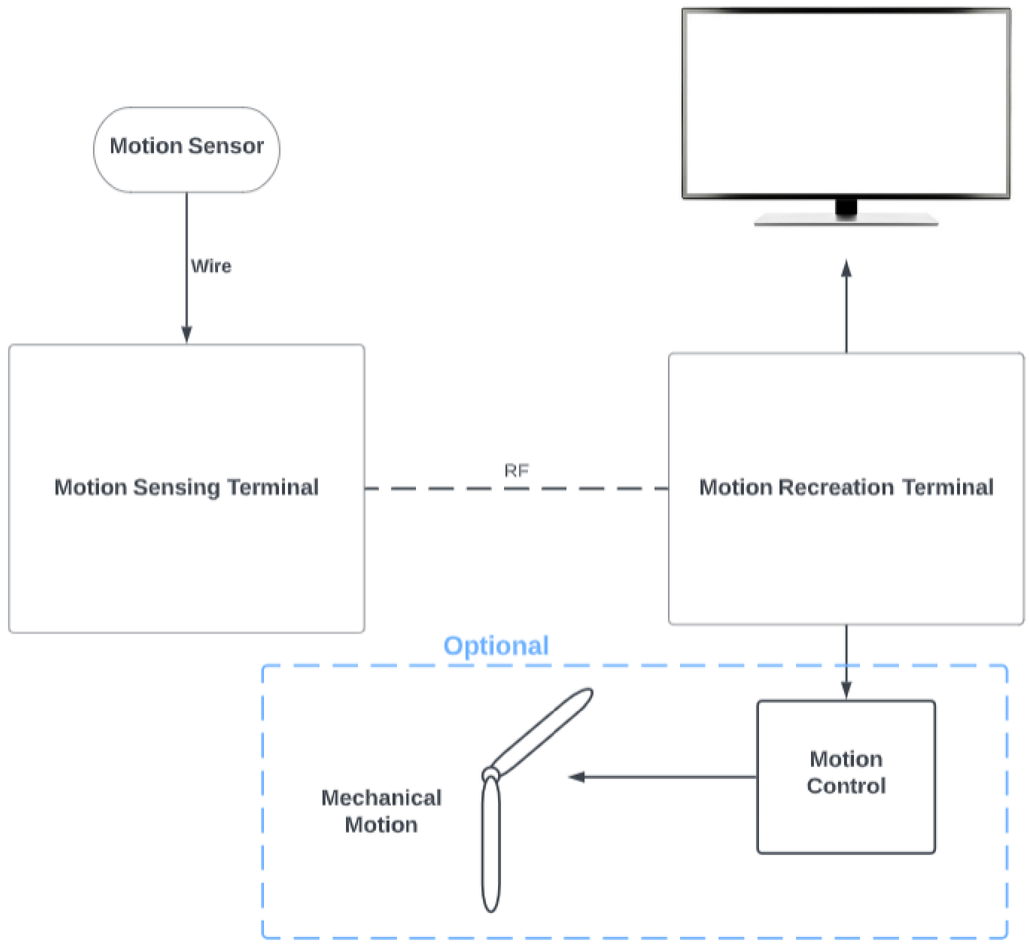
Aside from peer-to-peer communication, a long-range motion communication system can be useful in a variety of scenarios. In a classroom setting, whenever a Physics teacher wants to dig further into a relatively abstract concept, like lattice structure and electron concentration in materials, board and chalk and other variants are their only reliable helpers. However, it would be more engaging for both the lecturer and the learner to see a 3D presentation of the topic in question that's able to move and change at our commands. Likewise, a controller that's able to move extended robot arms can sometimes prove to be ineffective, as not everyone is acquainted with controller maneuvers. It will be much easier to understand and control if one is able to move the arm in real time with points of reference placed on limb joints that match the ones on the machine. Other utilities include but are not limited to workplace security, drone navigation, and smart home. All of which can and will be made simpler with a motion recognition and communication system.

Solution

We propose a duo-terminal system that reads motion data and sends the encoded information through RF communication to the other terminal which deciphers the data and reproduces the motion in real-time with 3D software simulation or mechanical integration like a motor. Built upon a previous project that clones movement data generated from MEMS sensor measurements to 3D animation, we will still work with discrete accelerometer and gyroscope measurements with appropriate sampling rate to ensure a seamless recreation even in a wireless setting.

Two PCBs are needed for each terminal. While most other components for this project will be printed to PCB, for the sake of flexibility of arrangement, IMUs, aka motion sensors, will preferably connect to the rest of the system via STEMMA QT or long wires. Unfettered from the confinement of circuit boards, IMUs in free space can adapt to more situations and diversify the motions they can output. A great example is the VR controllers that accompany most VR headsets.

Visual Aid

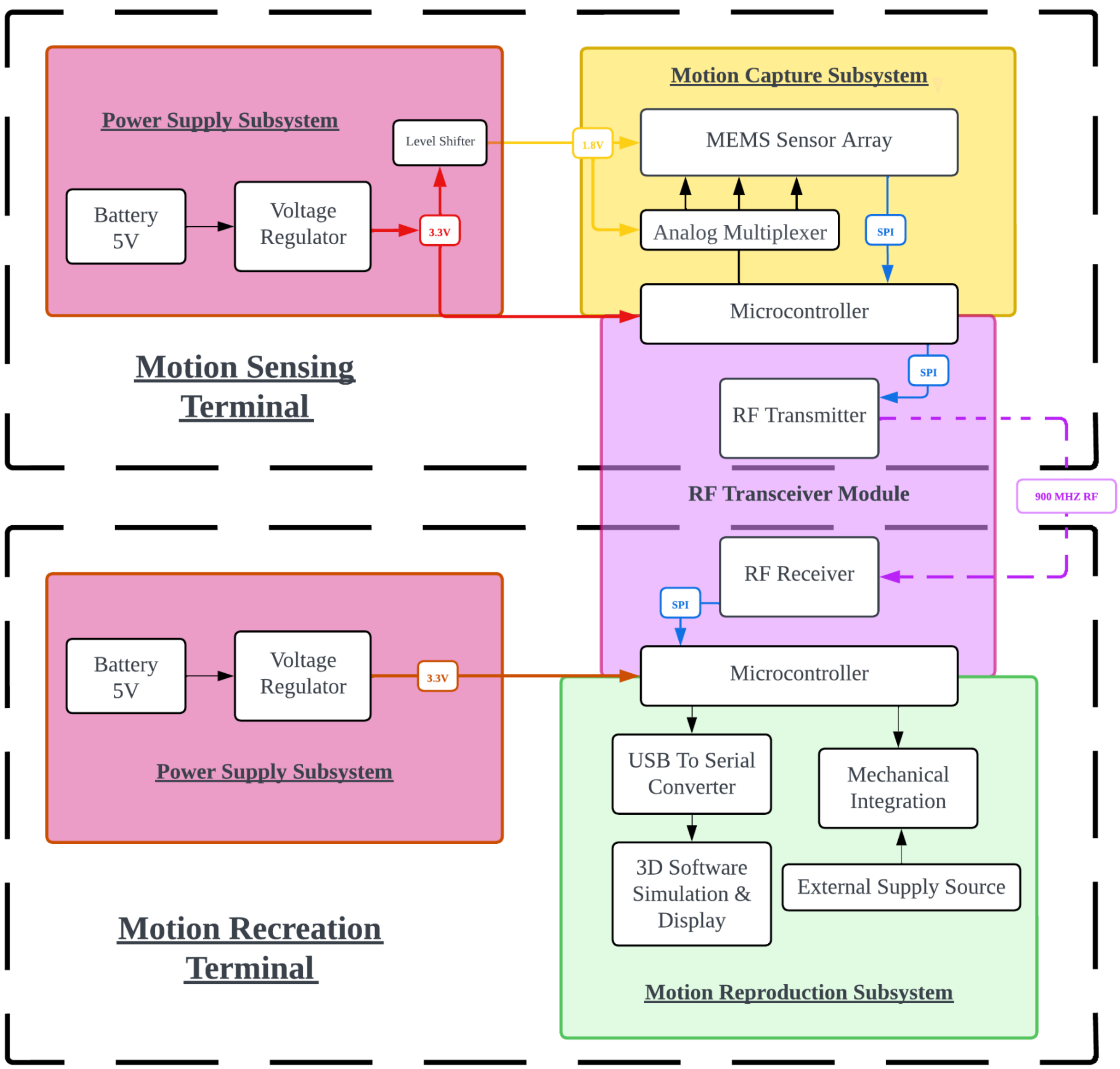

Motion Sensor
Wire
Motion Sensing Terminal
RF
Motion Recreation Terminal
Optional
Mechanical Motion
Motion Control

High-level requirements list:

- Positional and rotational motions are captured through MEMS sensors and converted to human–readable data. The MEMS sensor has 6 degrees of freedom. Three for the gyroscope to capture direction and three for the accelerometer to capture movement. For our project we will be connecting the sensors in series to the pcb via wires. This will allow us to move the sensors freely while sending all the data to the pcb.
- RF system is able to function properly and transmits the aforementioned motion data from one device to another at least 0.8-1 mile apart. The RF module will be built into the pcb and connected directly to the main microcontroller.
- The reproduction system at the receiving end is able to dutifully repeat the motion set at the transmitting end on the software end for the minimum success criterion. If met, a 3D printed and/or motor-controlled hardware system can be built to further explore the potential of the project. It will also be implemented with the same pcb design as the motion sensing. The only difference will be in the programming which turns it into a RF signal receiver.

Design

Block Diagram:



Subsystem Overview:

The system is divided into two remote terminals, with each terminal having its own power supply, motion capture or reproduction system, and RF module. The terminals communicate with each other via 900mhz RF waves.

At the transmitting end, measurements from the MEMS sensor array are taken and processed by an SPI or I2C-enabled microcontroller. The data is then packed into a 16 to 32 bit code with at least 4 bits for position and 4 bits for rotation for each three-dimensional axis, which is then wirelessly transmitted via an RFM69HCW transceiver with an external antenna connector. At the receiving end, the data sent through the transmitter is recovered by another RFM69HCW module connected to another SPI-enabled microcontroller. The microcontroller analyzes and unpacks the code to extract the information and prepares the data for respective motion recreation.

The motion reproduction subsystem is mainly software, which is realized by receiving data sent through the serial port from the microcontroller at the receiver's end, which is then decoded by the Unity 3D engine to animate a 3D model in a fashion similar to the previous project mentioned.

Subsystem Requirements:

POWER SUPPLY:

Both RF components and MEMS sensors require a voltage of 3.3V. We will use AA batteries for the power supply, but onboard LDO from selected microcontrollers will help us control the output voltage. Both terminals use the same form of power supply. The mechanical component needs separate external voltage sources for the motors, preferably batteries as well.

MOTION-CAPTURING SUBSYSTEM

This system consists of two or more IMUs that are in free space. The LSM6DSO32 6-DoF Accelerometer and Gyroscope IC will fulfill the need. Since we are aiming to use STEMMA QT as the connector, the I2C communication protocol is favored. This subsystem in particular will likely reuse some of the codes and concepts developed in a previous project that can be found here: https://wiki.illinois.edu/wiki/pages/viewpage.action?pageId=785286420 [1].

RF TRANSMITTER SUBSYSTEM

Measurements from the registers of LSM6DSO32 will be sent to the Arduino or an SPI and I2C-enabled microcontroller that processes the information and packs them into a 16 to 32 bit code with at least 4 bits for position and 4 bits for rotation for each three-dimensional axis. The code sent via SPI interface will be transmitted wirelessly through RFM69HCW transceiver with external Antenna connector.

RF RECEIVER SUBSYSTEM

Information sent through the transmitter will be recovered by another RFM69HCW module connected to another SPI-enabled microcontroller. The microcontroller will analyze and unpack the code to extract the information and prepare the data for respective motion recreation.

MOTION REPRODUCTION SUBSYSTEM

The motion reproduction subsystem ideally consists of two major parts – software and hardware. The software section is realized by receiving data sent through the serial port from the microcontroller at the receiver's end. The Unity 3D engine will decode the information and animate a 3D model in a fashion similar to the previous project done by Joe Luo mentioned above. (https://wiki.illinois.edu/wiki/pages/viewpage.action?pageId=785286420) [1] The hardware component consists of a mechanical integration that's able to recreate simple directional movements, like 3D printed structures or pulse-controlled continuous rotation servo motors (FS90R) that rotate on a 2D plane on a scale dependent on the degrees of rotation of MEMS sensors.

Tolerance Analysis

Admittedly, the power consumption of RF devices has always been a concern for radio module circuits. Since the motion update needs to be real-time, the radio modules should rarely enter idle or sleep modes, which may cause additional power consumption. To estimate the rough power consumption in the presence of RFM95W and additional IMUs, we take power ratings, idle current, and power supply into account. The master equation [3] for total consumed energy is :

$$ETotal = ESleep + EActive$$

From which, we deduct the model we use to calculate the power consumption.

$$ESleep = PSleep \cdot TSleep$$

$$EActive = EWU + Em + Eproc + EWUT + ETr + ER$$

Since we are mostly concerned with the energy for wake-up of the LoRa transceiver and energy of transmission or reception mode as explained above but not microcontroller ratings here, we can simplify the equation to

$$Etotal = EWUT + ETr + Er$$

The consumed energy EWUT during transceiver wake-up duration TWUT is given by:

$$EWU = PON(fMCU) \cdot TWU$$

The consumed energies of transmit and receive mode are expressed respectively as:

$$ETr = (PON(fMCU) + PTr) \cdot TTr \qquad \text{(Eq. 1)}$$

$$ER = (PON(fMCU) + PR) \cdot TR \qquad \text{(Eq. 2)}$$

Of which $Tr = Nbit \cdot Tbit$, where Nbit is transmitted bits and the duration of one transmission. To obtain the metrics for relevant parameters, we use the equation for bit-rate (in bits per second) as shown below:

$$Rbit = SF \cdot BW/2\text{^}SF \cdot CR \qquad \text{(Eq. 3)}$$

Where CR represents a controllable coding rates, and the rest are displayed in the tables for RFM9X below:

Part Number	Frequency Range	LoRa™ Parameters			
		Spreading Factor	Bandwidth	Effective Bitrate	Sensitivity
RFM92	860 - 1020 MHz	7 - 12	125 - 500 kHz	0.3 - 20 kbps	-137.5 dBm
RFM93	860 - 1020 MHz	7 - 9	125 - 500 kHz	1.7 - 20 kbps	-130 dBm

Symbol	Description	Conditions	Min	Typ	Max	Unit
IDDSL	Supply current in Sleep mode		-	0.1	1	uA
IDDIDLE	Supply current in Idle mode	RC oscillator enabled	-	1.5	-	uA
IDDST	Supply current in Standby mode	Crystal oscillator enabled	-	1.4	1.6	mA
IDDFS	Supply current in Synthesizer mode	FSRx	-	4.5	-	mA
IDDR	Supply current in Receive mode	*LnaBoost* = 00	-	10.5	-	mA
IDDT	Supply current in Transmit mode with impedance matching	RFOP = +20 dBm, on PA_BOOST RFOP = +17 dBm, on PA_BOOST RFOP = +13 dBm, on RFO pin RFOP = + 7 dBm, on RFO pin	- - - -	125 90 28 18	- - - -	mA mA mA mA

The crystal oscillator wake-up time for RFM9X is 250us. Substituting Eq. 3 into Eq. 1 and Eq. 2 and factoring desired parameters into the equation, we thereby obtain the energy consumption for a complete communication for 10 seconds as what is illustrated below:

$$ETr = (250us * 3.3V + 1.5mA*3.3V) * [10*300*1000/2^10*1/(4+4)] * 10 = 8.81J$$

Hence, the average power consumption for transmission is given by 8.81J/10s = 0.881W, which is reasonable and within the range of power we estimated for the radio module circuit. Same goes for the receiving terminal.

Another point of interest is whether the bit rate supplied by the chip is sufficient for our information to be properly transmitted. Our measurement demands six axes of data, spanning the three from the accelerometer and the three from the gyroscope. The accelerometer is selected to have a ±8g sensitivity. However, considering the limit of human motion and accessibility of motors in the hardware reconstruction, we limit the maximum acceleration transmitted to ±255m/s^2. The gyroscope, on the other hand, outputs degrees ranging from -180 to 180. As a result, we need at most 32 bits for each of the axes and one data header and footer, with a sampling rate of 1Hz, at least

$$\text{Total Bits} = 36 * (6+2) = 288 \text{ bits}$$

are needed for one successful transmission per second. The calculation from Eq.3 nets us with a bit rate of 366.21 bits/second, which is well above the needed transmission rates of data.

Ethics and Safety

When working on a project, it is important to consider the ethical and safety issues that may arise both during development and from the potential misuse of the project. Adhering to the IEEE Code of Ethics [2] and ACM Code of Ethics, which emphasize the importance of avoiding harm to others and upholding integrity, is crucial to avoiding ethical breaches. To ensure compliance, it is necessary to review relevant safety and regulatory standards such as state and federal regulations, industry standards, and campus policy. This project has been approved, which means the general purpose of this project is ethical and safe. However, more attention will be needed for each step in realizing the project. Additionally, potential safety concerns in the project should be identified and addressed in order to minimize any potential harm to users or others. For example, if the project involves handling sensitive personal information, steps must be taken to ensure the data is protected and secure. The project will involve motors in the motion recreation system, where injury could happen if the mechanical system is not placed in a secured place. We will make sure to keep the mechanical system in a secure location and only use it during mechanical testing. By considering these factors and taking appropriate precautions, it is possible to ensure the ethical and safe development of the project

References

1. Unity-enabled gesture motion capture (Joe Luo) - ECE 395 Advanced Digital Projects Lab - Illinois Wiki. [Online]. Available: https://wiki.illinois.edu/wiki/pages/viewpage.action?pageId=785286420. [Accessed: 09-Feb-2023].
2. “IEEE code of Ethics,” IEEE. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 09-Feb-2023].
3. Taoufik Bouguera, Jean-François Diouris, Jean-Jacques Chaillout, Randa Jaouadi, Guillaume Andrieux. Energy consumption model for sensor nodes based on LoRa and LoRaWAN. Sensors, 2018, 18 (7), pp.2104. ff10.3390/s18072104ff. ffhal-01828769f