

EM-Comm: Touch-based Communication via Modulated Electromagnetic Emissions

CHOUCHANG (JACK) YANG and ALANSON P. SAMPLE, Disney Research

Touch-based communication offers a direct and intuitive way for users to initiate and control data transfer when using tangible and ubiquitous interfaces. However, this requires that each interactive device be instrumented with a dedicated radio transmitter, which limits many applications. While not all devices have radio hardware, all devices do emit small amounts of electromagnetic noise in the form of EMI. We argue that if properly modulated these electromagnetic emissions can be used as an untapped communication channel capable of transmitting arbitrary data.

To enable this, a spread spectrum frequency shift keying modulation scheme has been developed to encode data onto the device's EMI. When the device is touched by a user, the data encoded EMI signal travels through their body and into our custom wrist band, consisting of a single op-amp and MCU. Our results show that we are able to turn electronic primitives such as LEDs, buttons, I/O lines, LCD screens, motors and power supplies into radio transmitters capable of touch communication. Effective data rates range from 5.8Kbps to 22 bit per image depending on the primitive used. To demonstrate this new communication technique, we develop several interactive experiences where users can retrieve complex information such as the function of buttons on a device, directions embedded into a LCD screen, and simplified device pairing. Ultimately, EM-Comm enables nearly any electronic device to be turned into a touch-based radio transmitter with only a software upgrade.

CCS Concepts: • **Human-centered computing** → **User interface management systems**;

Additional Key Words and Phrases: Touch Communication; Screen Reading; Electromagnetic Emissions; Spread Spectrum Frequency Shift Keying; Body Channel Communication; Wearable Computing

ACM Reference format:

Chouchang (Jack) Yang and Alanson P. Sample. 2017. EM-Comm: Touch-based Communication via Modulated Electromagnetic Emissions. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 3, Article 118 (September 2017), 24 pages. <https://doi.org/10.1145/3130984>

1 INTRODUCTION

Effective means of automatically understanding how we interact with and manipulate objects and devices in our daily lives has the potential to enable a wide range of tangible and ubiquitous user interfaces. Many research efforts have focused on observing object interaction detections by embedding sensing and computing devices into objects of interest, which then wirelessly report information back to a host computer for processing. While this works well for a small number of objects in a research setting, it does not easily scale due to the high per unit cost of the sensor nodes, requirement of manually instrumenting each object and issues related to battery maintenance. Furthermore object interaction is only part of the challenge, automatically understanding the state

Chouchang (Jack) Yang, Disney Research Pittsburgh, 4720 Forbes Ave, Pittsburgh, PA 15213; email: jack.yang@disneyresearch.com; Alanson Sample, Disney Research Pittsburgh, 4720 Forbes Ave, Pittsburgh, PA 15213; email: alanson.sample@disneyresearch.com .

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, or post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2017 Copyright held by the owner/author(s). Publication rights licensed to Association for Computing Machinery.

2474-9567/2017/9-ART118 \$15.00

<https://doi.org/10.1145/3130984>

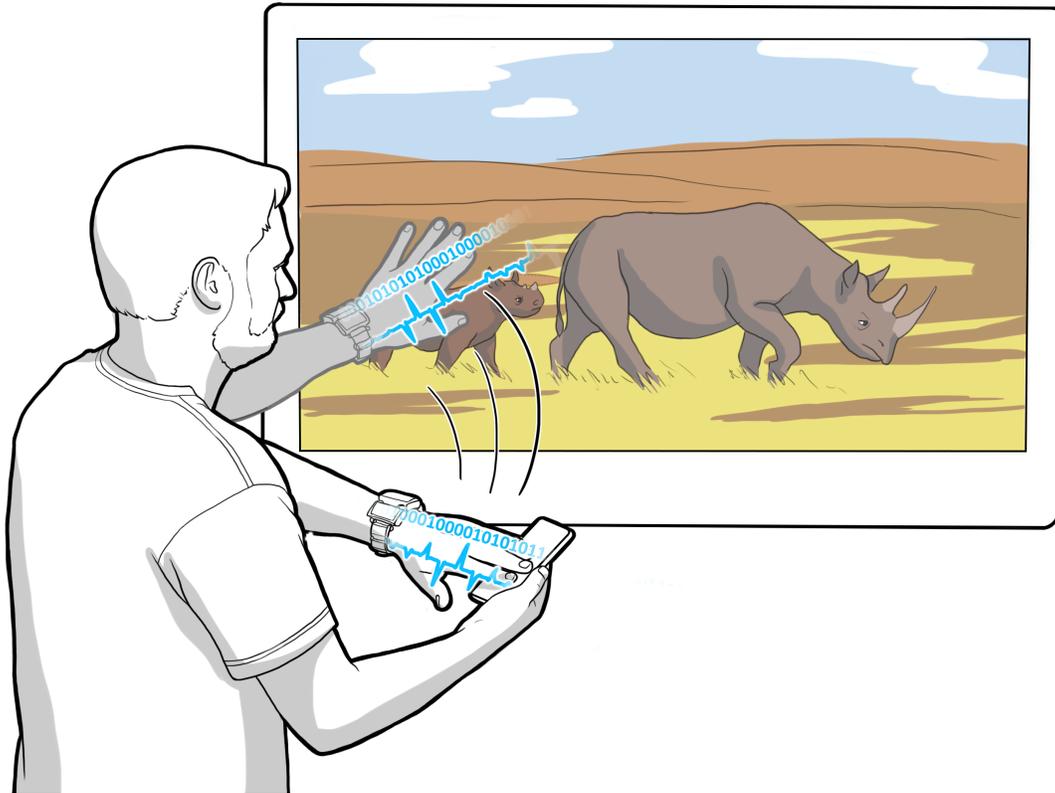


Fig. 1. The video playing on the public display encodes data into the electromagnetic noise generated by the LCD panel's row and column drivers. To retrieve this data the user touches the LCD screen, which transfers the data along his hand and into the EM-Comm wristband for decoding. He then initiates the transfer of the retrieved data to his smart phone by touching the phone.

of the device the user is interacting with provides a rich amount of information about the users implicit and explicit intentions [28].

Leveraging previous work on tag-less device interaction detection via electromagnetic sensing [18, 32, 33], we make the following observations: Nearly all electronic devices, regardless of complexity or intended use, are actively transmitting signals in the form of electromagnetic emissions (EME) also commonly referred to as electromagnetic interference (EMI). In this work we argue that these electromagnetic emissions represent an untapped communication channel, that if properly modulated can turn electronic primitives such as LEDs, buttons, I/O lines, LCD screens, motors and power supplies into radio transmitters capable of touch communication. All that is needed is software to encode base-band data into the EMI. No additional wireless hardware is required.

In order to exploit this capability we have created a simple wrist worn receiver consisting of a single op-amp and microcontroller, with an integrated analog to digital converter to capture and interpret the EMI data. By touching a device of interest, data encoded on top of the EMI (using our spread spectrum frequency shift keying modulation scheme) is transferred along the users body and into our smart watch receiver.

This core communication capability enables a number of interaction scenarios based on touch and physical contact. For instance Figure 1 shows a pictorial of a person transferring a movie trailer from a public display to their smart phone by “grabbing” the video from the screen and “placing” it onto their phone. Here the LCD panel emits EMI signals based on the operation of the pixel’s row and column drivers. By encoding baseband data as imperceptible variations in contrast in the video stream, the LCD panel transmits EMI data. With a data rate of 10kbps the movie’s meta-data along with a pointer to the video is transmitted along the users hand and into the EM-Comm smart watch. Then the user touches the smart-phone, which is already paired with the EM-Comm watch, to complete the transfer. Ultimately, EM-Comm allows nearly any electronic device to send arbitrary data payloads through touch, without the need for dedicated radios or hardware modification.

Our work makes the following contributions:

- The use of electromagnetic noise (i.e. EMI) as a carrier signal for data transmission
- Six electronic primitives that can be used for EMI data transmission including: I/O lines, LEDs, Buttons, LCD screens, Motors, and DC-DC power converters
- A Spread Spectrum Frequency Shift Keying modulation scheme for encoding and decoding EMI data
- Two methods for encoding EMI data into video streams that can then be transmitted by LCD screens
- A wearable smart watch architecture that receives EMI through the body and decodes EMI data in real time

2 BACKGROUND

Nearly all electronic devices produce some form of unintended electromagnetic emissions do to the normal operation of their circuitry. This is a fundamental outcome of Maxwell’s equations, which predict that any time-varying current will produce electromagnetic radiation. Typical circuit components that produce strong sources of EME include oscillators, digital clocks and I/O lines, switch mode power supplies and motors, which emit a large central tone and higher order harmonics.

In order to insure that these unintended emissions do not interfere with traditional radio transmission systems such as TV, Wi-Fi, and cellular services, governmental regulations have been adopted that strictly limit the magnitude of these emissions. In the U.S. the Federal Communication Commission (FCC) has two sets of standards [1]. One for conducted emissions between 1kHz - 30MHz (e.g. EM noise that propagates over power lines) [Part 15.107], and a second set for free space radiated emissions between 30MHz to 10GHz [Part 15.109]. These regulations set a hard limit on the amplitude of the EME, but not the form or “modulation” of these signals.

Therefore once a device has met FCC certification allowing it to be marketed and sold in the U.S. there are no restrictions on modulating the preexisting electromagnetic noise¹. For example, there are no limits on the types of images that can be shown on LCDs screens due to EME regulations. Since this data source does not change the magnitude of the emissions generated by the LCD hardware.

3 RELATED WORKS

At its core EM-Comm provides a touch-based (i.e. near contact) method for turning electronic devices into radio transmitters without the need for dedicated radio hardware. This method is suitable for new devices or existing devices, requiring only software modifications. When paired with the EM-Comm wristband receiver it opens up a wide variety of touch-based interaction usage scenarios where a devices identity, state information, and metadata can be transmitted to the user via touch. Related work can generally be placed in one of three

¹The FCC also set standards for low power, non-licensed transmitters operating at 9kHz and above [8]. The emission limits are generally more permissive then either the conducted or radiated emission limits and also do not restrict the modulation type; provided they do not interfere with sensitive radio communication systems such as aircraft radio navigation, radio astronomy, and radios used for search and rescue operations.

categories: 1) tagging systems where an object/device is augmented with dedicated radio hardware 2) Body Channel Communication where the signals are transmitted along the body often through touch, and 3) RF and EMI sensing methods for understanding gestures and users interactions.

3.1 Tagging

One standard approach to object interaction detection is to instrument each object of interest with battery powered sensor nodes that use traditional radios (such as Bluetooth or Wi-Fi) to stream data to a host computer to infer object interactions [10, 29, 31]. While this approach can deliver high fidelity streaming sensor data over distances of 10s of meters. It does not allow the user to explicitly control when and with which device to receive data from. Furthermore it also requires that each device be manually tagged with a relatively high per unit cost (tens of dollars), large size tag that will require manual recharging.

In order to reduce cost and increase the ease of deployment researchers have turned to battery free RFID tags that come in a sticker like form factor. These fixed function devices report their unique ID when energized and interrogated by an RFID reader. Near-field RFID tags have been successfully used to make tangible interfaces [21] and Phillips et. al. has developed a mobile RFID reader bracelets for detecting object interactions and ultimately activity inferencing [9, 27]. Likewise UHF RFID tags, which have a read range of up to 10 meters, have also been used for object interaction detection and tangible user interfaces [2, 19, 24, 30]. This reduces the need for a hand held or wearable reader by placing UHF RFID readers in the environment. While both these approaches reduce the burden of object / device instrumentation they still require manual tagging and the RFID tags are limited to transmitting their fixed ID as apposed to general-purpose data.

3.2 Body Channel Communication

Research on Body Channel Communication (BCC) has focused on using the human body as a medium to transmit data through [7, 25, 34]. Typically various forms of capacitive coupling are employed to transmit data between wearable BCC transceivers placed on the body, or between a wearable BCC transceivers and BCC node embedded in objects. While there are challenges related to ground effects BCC techniques offer the ability to not only detect human object interaction events, but also transmit arbitrary data between two [11, 12, 17, 22, 23, 35]. While this works well when dealing with a handful of well-instrumented devices in a controlled environment, the cost and complexity of adding dedicated BCC radios to all the electronic devices that we typically interact with has proven to be prohibitive at scale.

3.3 RF and EMI Sensing

The use of existing RF and EMI signals have been utilized in literature to enable a number of HCI applications [3, 13, 14, 26]. Cohn et. al. [5] used the human body as an antenna for receiving RF signals in the home. Here a small electrode was attached to the back of the neck and connected to a backpack-bounded spectrum analyzer. As the user moves around the home the system inferred user location within a home, as well as detect different gestures and continuous touch tracking along a wall. A later extension enabled free-space, whole body gestures by utilizing EM signals properties [4].

Recent work on EMI sensing has shown that it is possible to classify the category of the device a user is touching by measuring the electromagnetic signatures that it emits. Here the EMI signals emitted by the device pass through the users body to a wristband connected to a software define radio [18]. Follow on work has shown that it is possible to individually identify an instance of a device out of a given population using only the EMI signature [33]. However, in neither case was active data transmission explored.

The closest related work to EM-Comm is by Hessar et. al. which uses a fingerprint scanner to capacitively transmit data to a wrist band connected to a software define radio [16]. Here the fingerprint reader's Active

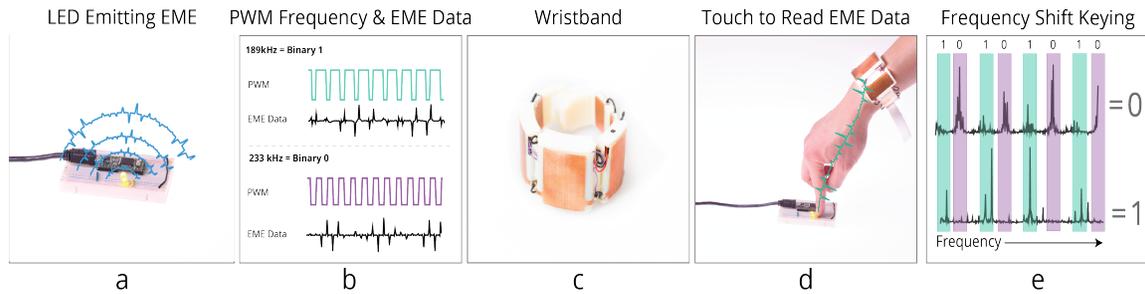


Fig. 2. System Overview: Panel (a) shows a microcontroller emitting electromagnetic noise due to the PWM signal outputted on a GPIO pin. In order to encode data into the EMI the system toggles between two different PWM frequencies, which generates two different EMI waveforms, as denoted in panel (b). The next panel (c) shows our custom designed EM-Comm receiver in wearable wristband form factor. When the user touches the microcontroller the EMI signal passes along their hand and into the wristband, panel (d). Once digitized, the EM-Comm receiver computes the FFT of the data and checks the frequency bins corresponding to a logical “0” in purple and a logical “1” in green, as shown in panel (e).

Capacitance Sensing scanner is turned on and off at a rate of up to 50 Hz to encode data. While this work nicely shows the value of sending data through touch and the resulting security implications as related to fingerprint readers, this capacitive coupling method does not scale to a wide range of devices.

4 SYSTEM OVERVIEW

This section provides a general overview of the process of encoding data onto the electromagnetic emissions produced by a device, transferring that signal through touch to the EM-Comm wristband, and the general process for decoding the data. Subsequent sections provide a deeper exploration of the six circuit primitives we have developed that can be used to transmit EME data and the algorithms needed to encode and decode that data, along with applications of this technology. From physics we know that any time varying current will produce some amount of electromagnetic radiation. For the purposes of this work we are referring to non-ionizing radiation in the form of low frequency radio waves, below 1 MHz. The key concept is that electronic primitives such as digital I/O lines and clocks, oscillators, DC to DC power converters and motors all give off relatively strong amounts of electromagnetic emissions at the same frequency as the oscillating current.

Lets consider the microcontroller development board shown in Figure 2a, consisting of an ARM Cortex M4 and supporting circuitry. This unmodified piece of hardware can be programmed to emit EME at specific frequencies by toggling one of its GPIO pins with a digital timer. For this example we choose an unloaded I/O pin but, as will be shown later, circuit elements can also be driven in a manor that produces usable EME. In order to encode data we use a Spread Spectrum Frequency Shift Keying (SS-FSK) modulation scheme that spreads the data across the frequency domain thereby making it less sensitive to jamming from other sources of electromagnetic interference. For example, Figure 2b shows two Pulse Width Modulated (PWM) wave forms generated by the MCU and the resulting EME signals in the time domain. The PWM signal switches between 0V and 3.3V. Selecting a rate of 233kHz with a 50% duty cycle, denotes a logical “0” and a PWM signal of 189kHz with a duty cycle of 50%, denotes a logical “1”. The resulting EME signals can be seen as the derivative of the digital pulses, which makes sense since the rising and falling edges contain most of the high-frequency components (i.e. fastest time varying currents). By switching between these two PWM frequencies it is possible to modulate the EME to encode arbitrary data. While this particular encoding method works well for most circuit primitives more sophisticated encoding methods are needed for LCD screens as will be presented in later sections.

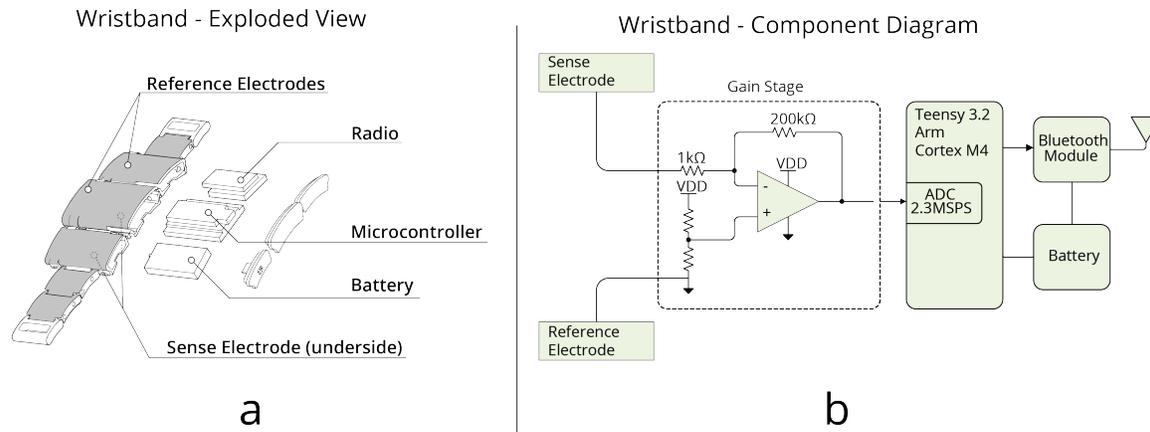


Fig. 3. Panel (a) shows the mechanical design of the EM-Comm wristband consisting of a 3D printed body with top and bottom electrodes for coupling to the user. Along with the microcontroller PCB, battery and Bluetooth radio module. Panel (b) shows a schematic of the EM-Comm receiver consisting of an inverting amplifier for signal conditioning, a Cortex M4 microcontroller with 2.3MSPS ADC, as well as the battery and radio.

To receive the EME data signals emitted by the microcontroller we have developed a self-contained wristband as shown in Figure 2c. The EM-Comm wristband consists of top and bottom electrodes for coupling to the skin, an analog front-end for signal conditioning and gain, an on-board MCU with 2.3 MHz ADC for digitizing the EME signal, along with an external Bluetooth module and battery.

Figure 2d shows a user wearing the EM-Comm wristband and touching the top of the microcontroller on the development kit. The EME signal encoded with data is coupled into the users finger and travels along the hand and into the wristband where it is digitized via the ADC. In order to decode the EME data the MCU on the wristband converts the signal into the frequency domain as seen in Figure 2e. Due to our Spread Spectrum Frequency Shift Keying modulation scheme, data appears at multiple frequency bins in the frequency domain. By looking for peaks in the “0s” bins (denoted in blue) or the “1s” bins denoted in red it is possible to robustly decode the EME data. Typical transfer rates are on the order of 10 kbps.

4.1 EM-Comm Wristband Structure

We proposed a wristband system as a receiver to decode the data by capturing EM signals. The wristband consists of three main components: Teensy 3.2, Bluetooth, and an OPAMP front end circuit as illustrated in Figure 3a. In Figure 3b, the EM signals from the sensor port are first amplified via an OPAMP with a gain of 200 and then are sampled at 2.3 MSPS by using the 12 bit ADC in Teensy 3.2 MCU where the CPU uses ARM cortex M4 with the overclock 144 MHz setting. The OPAMP chip we used in here is TI THS4221 which is a low distortion and high speed operational amplifier with 230 MHz -3dB gain-bandwidth product. Since the 2.3 MSPS sampling rate has 1.15 MHz baseband bandwidth, the maximal gain provided by TI THS4221 is up to 200. In here, we set the gain to the maximal 200 such that the EM signals can be easily detected when users make contact with electronic objects. The maximal input bias current for TI THS4221 is $0.9 \mu\text{A}$. The VDD in Figure 3b is 3.3 V and reference voltage is 1.6V which gave linear region for output voltage from 0V to 3.3V. Our resistance configuration guaranteed

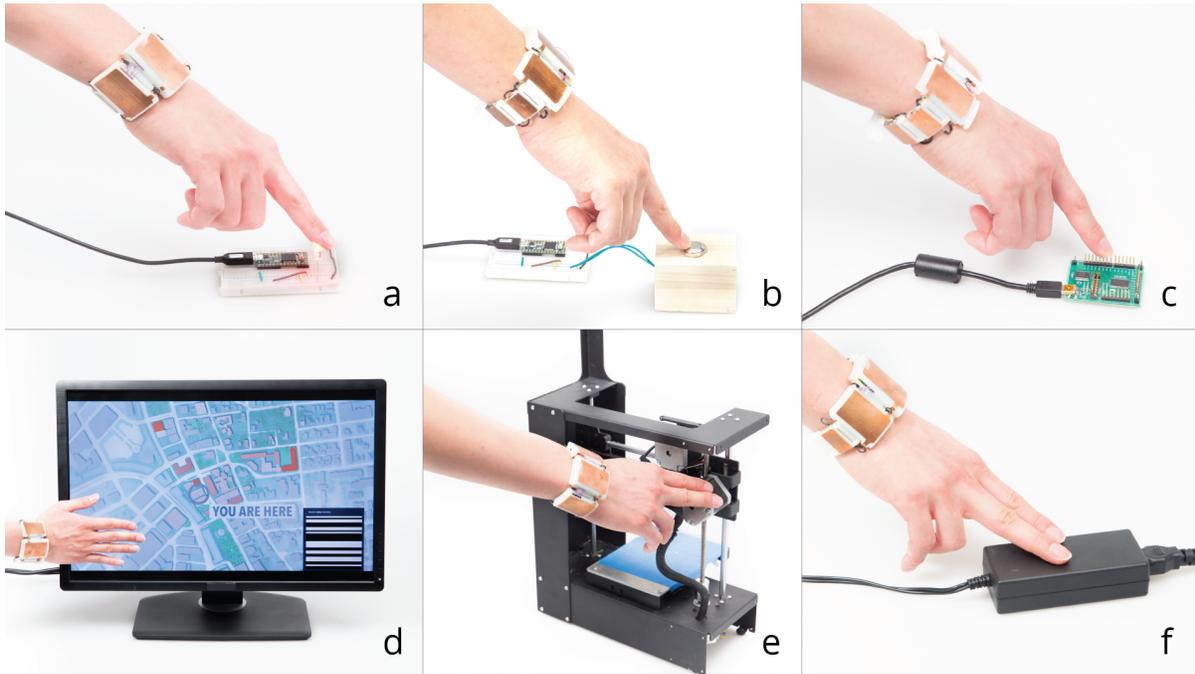


Fig. 4. Electronic primitives such as LEDs (a), Buttons (b), I/O lines (c), LCD Screens (d), Motors (e), and Power Supplies (f) emit small amounts of electromagnetic noise that when properly modulated can be used to transmit arbitrary data.

EME input-bias current and input voltage are within TI THS4221 operating ranges. The wristband has utilized a similar electrode design appeared in [18] where the reference electrode is outer surfaces while the sense electrode is inner surface contacted with the user skin. The electrodes are made by using copper foil. Our EM-Comm wristband does not require to have common ground with any electronic device for reading EME data since we have adopted high gain amplifier to enhance EM signals quality.

After receiving the EM signals, the MCU decodes the EM data by processing the incoming data samples and then sends the decoded result via Bluetooth connection back to the host computer. The Bluetooth module used here is the RN41 Bluetooth chip and the throughput is set at 921 Kbps. Although one can also use BL link connection to send the raw ADC samples back to host computer for further processing, Bluetooth is not able to retrieve all the raw sampled data since the 2.3 MSPS sample rate with 2 bytes for each sample is beyond Bluetooth throughput. Thus, we implement an on-board decoding algorithm in the EM-Comm wristband to extract the EME data information and send back the results to host computer via Bluetooth for providing further interaction services. Since the EM-Comm data throughput only consumes a small portion of BL link, we also let the remaining BL throughput send the raw sampled EM signals which contain packet data for optional debugging purposes and performance evaluation. Generally, the EM-Comm wristband is a stand-alone battery powered system which can decode all the encoding EM signals and use the decoded information for further service requests.

5 PRIMITIVES-I: LED, BUTTONS AND I/O LINES

All electronic devices emit some amount of electromagnetic noise. As long as it is possible to modulate these emissions between two different states it is possible to encode data into the EME in order to transmit data. In the primitives section, we present three different data transfer mechanisms using six common electronic primitives

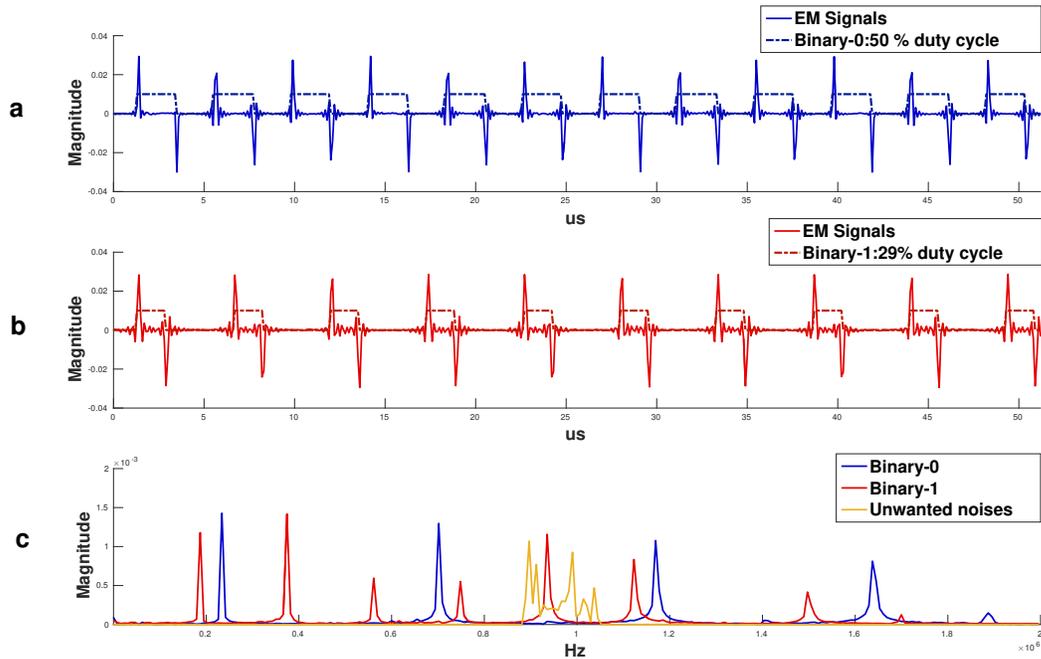


Fig. 5. Example time domain and resulting frequency domain plots of the Spread Spectrum Frequency Shift Keying scheme used to encode data into the electromagnetic noise by generated the electronic primitives.

that are used in our daily lives. They are 1) LEDs, 2) Buttons, 3) I/O lines, 4) LCD Screens, 5) Motors, and 6) Power Supplies. Here, we first introduce the microcontrollers-based primitive: LEDs, buttons, and I/O lines which are peripheral devices that can be controlled by microcontrollers (MCU) to inject the data for EM-Comm systems as illustrated in Figure 4a, b and c.

LEDs are one of the most common electronic devices in daily life, and their uses range from lighting to indicator. In the case of indicator LEDs, it is often found that they are connected directly to MCU GPIO pins. Similarly, buttons are likewise often found connected to GPIO for gaining user input. Buttons, unlike LED indicators that use the output pins of MCUs, are connected to a pull-up resistor at the input pin of MCUs. The MCUs read the voltage here to identify whether a click action has been made. For example, when the button is not pressed, the default high voltage appears at the input pin. By pressing the button, the input pin is shorted to ground such that the voltage then becomes low. By detecting the presence of the low voltage, the MCUs can respond to the pressed action. I/O lines can be viewed as a line extension from the MCU's GPIO pins. The I/O lines usually are used to connect to peripheral devices such as LEDs and buttons. The I/O line can vary the amplitude to provide control and communication purposes. Therefore, we can leverage the use of digital I/O pins to embed data while providing the original function. The ubiquity of LEDs and buttons, as well as their direct connection to the the MCU I/O pins make them uniquely suited to data encoding, with one of the key advantages being that the wires lead directly from the MCU to the exterior of electronic devices. Thus, this permits easy use and extended range.

In one example of the proposed system, a digital output pin can turn on LEDs by setting the voltage high. Human eyes are not able to detect blinking if the switching frequency of this is faster than human perception.

Hence, LEDs are able to provide their original functionality as indicators, while additionally sending information for EME based communication. For buttons, using AVR-based MCUs, the GPIO pins can directly read the voltage status while also providing output on the same pin. We then can directly embed data while still reading the voltage status at the same time. For non AVR-based MCUs, we leverage the property that the GPIO pin is able to switch between input and output states within several clock cycles. By applying time multiplexing, the MCUs can switch between using the GPIO pin as input and output repeatedly, thus allowing data transfer in addition to detecting button presses.

Injecting EM signals in buttons also provide a notification in terms of button status for the EM-Comm wristband. When the button is pressed, the voltage is pulled low to ground and so the EM signal emission is also stopped. Then, after releasing the button, the EM signal suddenly reappears. When the wristband observes the fact that the EM signals have abruptly disappeared and then re-appeared, the receivers know the users have triggered the button's press action. In addition, the disappearing and re-appearing rate of pushing a button is different from putting a hand on the device and then taking it away, so we can use the varying rate to precisely detect the button touching action. In addition, we can also apply the opposite concept to enhance the button detection where a wrist band only detects a legitimate button push action when the click action corresponds to the desired number of state changes.

As mentioned, buttons, LED peripherals, and I/O lines are controlled by the MCU digital IO pins. Generally, we can choose two different frequencies, f_1 and f_2 , by choosing two different values of the signal period to represent binary data "1" and "0", where in both cases the switching frequency is faster than human visual perception. The frequencies of the signal's harmonics are dictated by the period, while the duty cycle sets the energy distribution in each harmonic frequency bin. In the next section, we illustrate how to design the desired frequency distribution to maximize the performance.

5.1 EME Spread Spectrum Frequency Shift Keying Data Encoding

To encode binary data, we propose a Spread Spectrum Frequency Shift Keying (SS-FSK) method that uses two different Pulse Width Modulation (PWM) pulse trains to encode data onto the EME signals. The sender generates a sequence of EM signals appearing every T_1 seconds to represent binary data "1" while using another EM pulse sequence appearing at every T_2 seconds to represent binary data "0". These two different pulse trains generates EME signal with different frequency distribution. Thus when converted in the frequency domain, with the FFT operation, the receiver can decode the incoming binary data based on the amount of energy in a given frequency bin.

5.1.1 Frequency Distribution Design. To have robust performance, we should have distinct frequency distributions between the two signal waveforms as binary data "1" and "0" such that the frequency distributions of both EM pulse sequences should not overlap. Thus, there is a system design requirement for choosing the appropriate time period T so as to have distinct waveforms as well as non-overlapping frequency distribution. Generally, an impulse sequence with time period T has a harmonic frequency distribution with components appearing every $F = 1/T$ [15]. Also, a digital binary waveform produces two different EM pulses in terms of its polarity. Take the Figure 5a as example, when a digital waveform switches voltage from 0V to 3.3 V, it generates a positive EM impulse response. Similarly, when the voltage is changed from 3.3 to 0V, it emits a negative EM impulse response. The blue dot line in Figure 5a is a digital waveform in GPIO pins by setting a PWM waveform with 50% duty cycle and time period $T = 4.3\mu s$, while the blue line is the corresponding EME RF signals emitted from the GPIO pins. Figure 5b applies a PWM signal with with 29% duty cycle and time period $T = 5.3\mu s$. The different PWM configuration in terms of time period and duty cycle made both have different frequency distribution as illustrated in Figure 5c. To further illustrate how the frequency distribution depends on the time period as well as duty cycle, we denote the positive EM impulse response and negative EM impulse response as $p(t)$ and

$n(t)$ respectively while using T to denote the frequency of PWM and α for the percentage of PWM as well as duty cycle. One should also notice that $n(t) = -p(t)$ due to the same magnitude of voltage change from low to high and high to low. Then, any EM signal emission through periodically varying digital waveform $x(t)$ and its frequency distribution $X(f) = FFT\{x(t)\}$ can be described as below

$$x(t) : p(t) + n(t - \alpha T) \xleftrightarrow{\text{FFT}} X(f) : P(f) - P(f)\exp(-j2\pi f \alpha T) \quad (1)$$

Where $X(f)$ is the Fourier transform of $x(t)$. The equation (1) can predict the frequency distribution. For example, when $\alpha = 50\%$, the $f = 1/T, 3/T, 5/T, \dots, (2L+1)/T$ in $\exp(-j2\pi f \alpha T)$ will result in -1 while $f = 2/T, 4/T, \dots, 2L/T$ will make $\exp(-j2\pi f \alpha T) = 1$. Hence, in Figure 5(a), the even frequency bins of PWM cycle 233 kHz will have no energy distribution while only odd frequency bins of 233 kHz cycle can have energy due to using 50% duty cycles. One can notice that there is a small energy remaining at $233 \text{ kHz} \times 8 = 1.86 \text{ MHz}$. This is because a few of the negative EM pulses may have smaller magnitude as compared to the positive EM pulses. Figure (5)b shows the PWM with corresponding EM signal when the frequency rate is 189 kHz with 29% duty cycle. The $\alpha = 29\%$ will not induce $\exp(-j2\pi f \alpha T)$ as 0 for most even harmonic frequency such as $f = 2/T, 4/T, \dots, 2L/T$. One can see the 7th harmonic frequency bin at 1.32 MHz has no energy, and this is because having $\alpha = 0.29$ results in the smallest magnitude distribution at that frequency bin. In short, the time period T dictates the harmonic frequency distribution while the duty cycle, α , determines how the energy is distributed among those harmonic frequency bins. Generally, 50% has the smallest number of harmonic frequency bins. In our case, there are five harmonic frequency bins for $\alpha = 50\%$ while there are 8 harmonic bins for a 29% duty cycle. Overall, we can use equation (1) to design the desirable frequency distribution by choosing corresponding duty cycle and time duration.

5.1.2 Guard Interval and Preamble. We now can embed data as binary 1 and 0 by producing a sequence of waveforms with different duty cycles as well as time periods. By checking predefined energy distributions in predefined frequency bins, the decoder can decode EM signals into binary data. To further assist the decoder in knowing the boundary of each sequence of EM pulses, we add a guard interval for time synchronization between each of the EM sequences. In the guard interval period, the EM transmitter will not generate any EM pulses to assist the decoder later in knowing the boundary. Additionally, the EM transmitter will first generate a predefined data format as a preamble to let the decoder know the starting point. In our system, we use '1111' as the preamble. Although we use a PWM digital waveform as an illustration here, one can also choose the corresponding binary stream (i.e. 000111000...) in any digital bus such as UART or GPIO to construct the desired signal waveform. The overall frequency distribution can be constructed by using the equation (1).

5.2 Spread Spectrum Frequency Shift Keying Decoder

To decode the EM signals, the receiver first needs to decide the boundary of each EM pulse sequence. Since the EM transmitter will use on-and-off states to generate guard interval, the receiver can use a sliding FFT window to search the boundary. For example, consider a receiver first using an N size of the FFT to transform the time domain EM impulse sequence into the frequency domain. Then, by sliding the FFT window along the time domain to search for the maximal signal in the frequency distribution, the receiver can identify the boundary for further data decoding.

5.2.1 Unwanted EM Signals Tolerance. Once the boundary is found, the receiver can start to interpret the frequency distribution as binary data 1 and 0. Then, the receiver can check all predefined harmonic frequency bins for both 233 kHz and 189 kHz first as shown earlier to select the appropriate frequency bins for further processing. The appropriate frequency bin is examined by checking whether the pulse appeared in those predefined frequency bins. Since we have adopted a frequency harmonic property to generate a Spread Spectrum Frequency Shift Keying modulation scheme, the wide band harmonic frequency bins can provide noise tolerance. For example, in

Figure 5c, the unwanted EM noise distorts the predefined waveform at the harmonic frequency bin of 935 kHz where the highest peak should only exist at one point while the local area has other peaks from unwanted EM noise. Thus, the harmonic frequency bin at 935 kHz is not viewed as a legitimate frequency bin to be selected, because the receiver has noticed that there are some other EM signals overlapping in that frequency bin. After extracting all the legitimate harmonic frequency bins, the receiver combines all the magnitudes for the same harmonic frequency bins. If the total energy at 189 kHz harmonic is larger than 233 kHz, then the decoder will view this waveform as representing bit 1. Then, by decoding all bit data and preambles, the receiver is able to truncate the bit sequence to the corresponding correct data format. While one of the harmonic frequency bins at 935 kHz in the binary "0" waveform is discarded by the receiver due to the unwanted noise, the other harmonic frequency bins can still assist the receiver to decode the data. Thus, our proposed Spread Spectrum FSK coding technique can provide robust data transmission against ambient EM noise.

Table 1. BER performance for Peripheral Devices based on different read range and data rate

Data Rate(bps)	LED		Button		I/O Line	
	WH	NWH	WH	NWH	WH	NWH
5.8K	9.1 %	21.4 %	12.4 %	31.2 %	7.3%	25.2%
1.4K	3.4 %	10.2 %	6.5 %	14.7 %	2.1%	8.4%
384	0 %	1.6 %	0.3 %	2.5 %	0 %	1.2 %
67	0 %	0 %	0 %	0 %	0 %	0 %

5.3 Performance

To further test the LED and button, we used a Teensy 3.2 MCU connected the digital output to either an LED, or a button, where in each case they are connected through a 5 cm long wire. For I/O line performance, we used a TI MSP430 development board which contains multiple I/O line as development interface. Here we denoted the "wristband hand (WH)" as the user's hand wearing the EM-Comm wristband while using "non-wristband hand (NWH)" for the other free hand. For example, if a user wore the wristband in his/her left hand, the wristband hand is left hand while the non-wristband hand is right.

In our system, the data rate bit/s can be calculated by $1/(\text{Data Time} + \text{Guard Interval})$. The data time is the total execution time for decoding a single bit, which includes the sample collection time and N-point FFT window execution time for decoding data. We designed the guard interval time to be the same as data time. This guarantees the decoder has enough calculation time to slide the window for finding out the boundary of each bit transmission.

We set the FFT window size as 64, 256, 1024, and 4096 differently such that the corresponding data rate is 5.8K, 1.4K, 384, and 67 bit per second respectively. The BER performance is evaluated by collecting at least 1000 bits for each configuration. We let a user randomly touch the object for four rounds. In each run, the user touched the objects such as LED, button, and a I/O line for a few seconds long to have more than 250 bits EME data. Based on this goal, a user touch a object for around 1 second when the data rate is configured as 5.8K/1.4K/384 bps. For the data rate 67 bps, we let the user touched the object for at least four seconds to have enough data for evaluation. The overall performance is illustrated in Table 1. When the FFT window size is 64, it has the fastest data rate of 5.8 kbps for transmission. However, it also has highest bit error rate (BER). When increasing the FFT window size, the data rate is reduced due to consuming more time for processing larger FFT window and having the transmitter sending a much longer EM sequence for each bit transition. However, increasing the FFT window size also decreases the BER. The idea behind this is that when we increase the FFT window size, the number of EM pulses for each FFT window also increases. Generally, the total energy of EM signals in frequency distribution is based on the number of EM pulses in each FFT window. Thus, when increasing FFT window size, the receiver

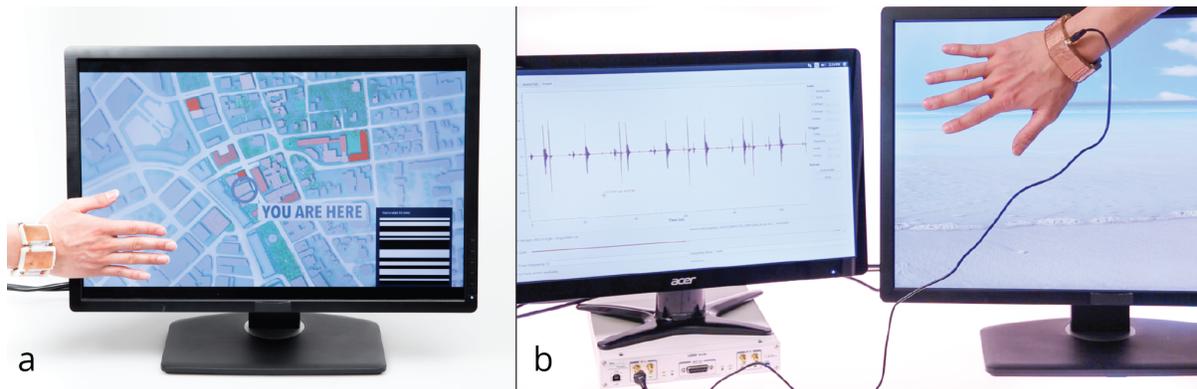


Fig. 6. Two different encoding approaches are introduced in LCD primitive. (a) Visible data encoding approach used a sequence of black and white line to encode data and a user wearing EM-Comm wristband can extract the data when touching the screen. (b) Invisible data encoding approach used the slight difference of the brightness to encode data and a user wearing USRP-based wristband can obtain the hidden data when touching the screen.

has received more EME signals to decode binary data 1 and 0. Thus, there is a trade-off between the data rate and BER. In here, FFT window size 1024 is recommended to be used for our application, since it has low BER and the 384 bps is able to afford most information retrieval service such as IP address and object status. Also, given a 20 bits packet size (16 bits payload + 4 bits preamble), the average packet loss under 1024 FFT window size configuration for LED, Button and I/O line is within 2% when using wristband hand to touch the above objects.

While in here we only consider binary frequency shift keying, in future we consider to use multiple frequency shift keying to maintain data rate while still using large window size to achieve low BER performance as well as low packet loss rate. Besides, the hardware improvement from faster CPU and more sensitive EM wristband design can also be made in future to enhance data throughput and quality for EMI communication.

6 PRIMITIVES-II: LCD SCREEN

A regular LCD or TV screen can be used for data communication purposes by modulating its EM signals. By simply modifying an image with a certain pattern, a user can obtain information directly when touching the screen. In the LCD primitive, we will demonstrate two different data encoding approaches as illustrated in Figure 6. They are visible and invisible data encoding techniques. Figure 6a illustrates a visible data encoding approach where a visible coding pattern used a portion of the screen to transmit information and the stand-alone EM-Comm wristband is used to decode the result. The Figure 6b shows an invisible data encoding approach where the screen can transmit the information through the change of brightness such that human eyes cannot detect the data information. To decode the hidden information, a grounded hardware X310 USRP is used here which can sense slight difference of EM signals between different brightness. In next section, we will describe each coding and decoding approach in details.

6.1 Embedding Visible Data in LCD Screen

To further describe how we encode data through regular LCD or TV screens, we start by describing the LCD screen design first. A panel LCD screen uses an array of pixels to display images or videos. Each pixel contains three different color channels indicating red, green, and blue (R, G, B) colors. By combining different levels of the red, green, and blue channels, an LED screen is able to present various colors. An LCD screen presents images by first selecting a column of pixel information and then updating each pixel in the column. Each pixel is driven by

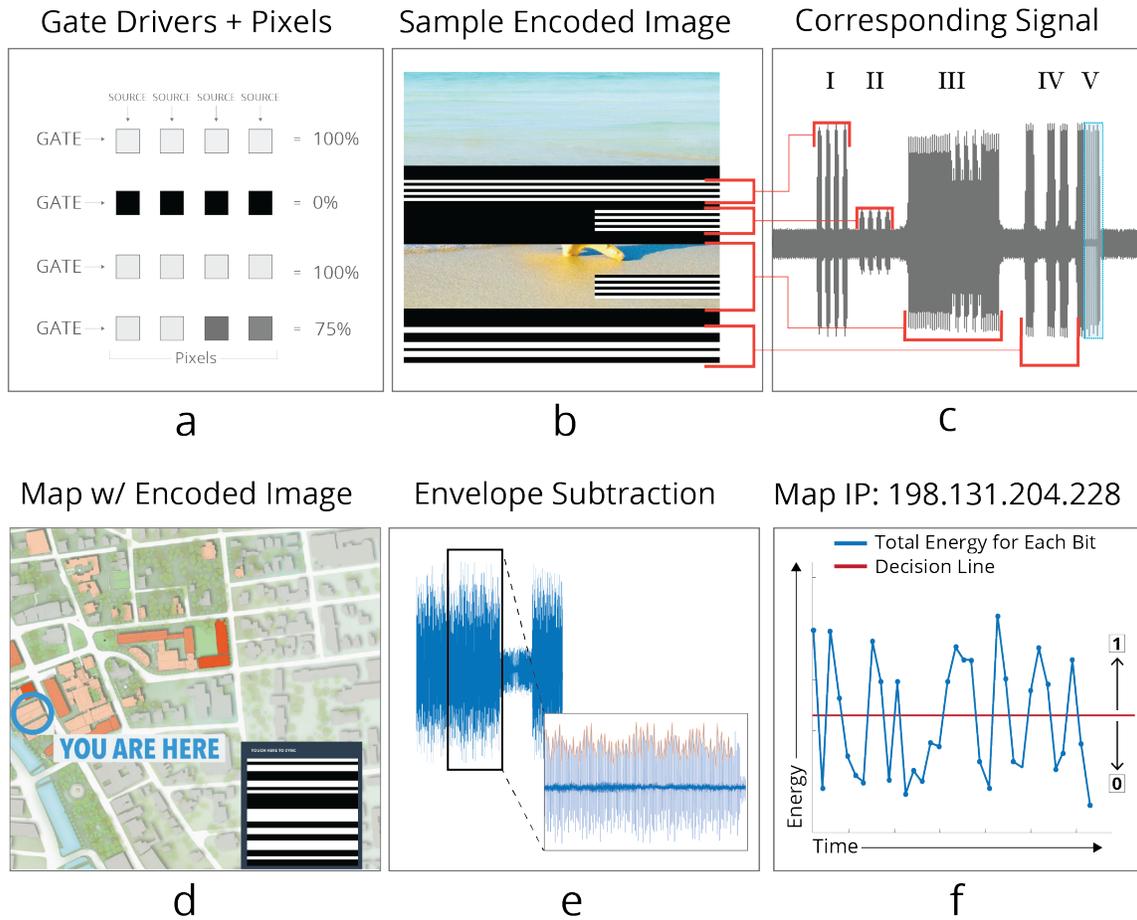


Fig. 7. Panels a, b, and c shows the underline principals of how row brightness effects the electromagnetic noise generated by LCD panels. Panels d, e, and f shows how “visible” data encoding can be used to encode data into the EMI and how that data is processed once received by the EM-Comm wristband.

a source and gate IC. The intensity of colors in the R, G, B channel is relative to the applied voltage magnitude. For example, if an R channel has a maximal magnitude of 255, then the source driver will send the maximal voltage around of 5V to that pixel. A gate driver usually uses more than 20 V as a transition to turn on and off the transistors in the selected column [6]. Thus, the EM signals we are interested in are generated by the gate driver due to its strong signal strength. Moreover, we found that the gate drivers drive different voltages based on the total pixel density in a row. Figure 7a illustrates that the gate driver enables a column of pixels in an array to be selected for changing information and thus it generates the corresponding EM signals. For example, if all pixels driven by source drivers are white as illustrated in Figure 7a, first row, then the total brightness of all pixels in the same row have the highest brightness and produce a gate driver with the strongest EM signals. If all the pixels are black in color, as in the second row, the gate driver can use the smallest voltage to enable this

row. Thus, the EM signal is at its lowest. If pixels in the same row have two white colors and two gray colors as shown in Figure 7a's last row, the total brightness combination therefore is 75 % of the whole brightness, thus the EM signal strength will follow this proportion. Since the EM signals from LCD screen is near field emission, a user can receive the same EM signals while touching in different areas of an LCD screen. Namely, the received information are not depending on touch location. The information only depends on the variation of EM signals across time made by each row of pixels.

Generally, the EM signals emitted from LCD screens have three main parts. The first is the EM signals from source drivers, which have very small magnitude and the second component is the EM signals from gate drivers, which, as mentioned, have the largest EM emission due to the high voltage changes in enabling a column of pixels. Finally, the last component of EM signals are the time synchronization control signals, called as HSYNC which appear after all the pixels are updated in each frame.

We use a Dell LCD screen 1920x1200 (Model:U2413F) as an example of illustrating how we can encode data using an image. As Figure 7b illustrates, black and white bars are displayed in the image while Figure 7c presents the corresponding EM signals along the time axis; the EM signals are measured by using a USRP X310 with 100 MSPS sampling rate. There are five different zones labeled from I to V at panel (c) for example purposes. In zone I, the Figure 7b first presents whole screen length black and white bars repeatedly four times, resulting in the largest signal amplitude variation as shown in Figure 7c where there are four individual groups of the strongest EM signals. In zone II, when only half screen width rows are used to represent white bars as shown in panel(b), the amplitude variation of EM signals illustrated in panel (c) still preserves the brightness variation but with much smaller amplitude. In Zone III at panel (b), the black and white bars share a portion of each row with a regular image. In Zone IV, if we choose different bar widths in terms of black and white at panel (b), the EM signals will follow the same order in terms of time duration. Finally, in Zone V, the EM signals are from the HSYNC control signals, which are not related to pixel information. The HSYNC (Horizontal Synchronization) signals are timing control signals and used for frame synchronization [6]. The HSYNC tells the screen that the current frame is done and to let all source and gate drivers be initialized for the next frame. The EM signals from HSYNC have longer time duration between each EM impulse compared to the EM signals from the gate driver. Namely, the EM pulses from HSYNC are more loosely packed in time while the EM pules from the gate driver are packed more closely to each other. The HSYNC can be used as an indicator to known where the data embedding area is, since all the EM signals appeared with precise timing control and their corresponding order is: EM signals from HSYNC, source, and then gate drivers.

6.1.1 Decoding Visible Data in EM-Comm Wristband. We have illustrated that there is a relation between pixels and EM signals. Now, we use Figure 7d as example to show how we can encode data in a regular image. Each bit here uses 350x10 pixels as black and white bar to encode "0" and "1" respectively. The 350 x 10 pixel blocks occupy a 350/1920 portion of each row and 10/1200 portion of each column in screen. These ratios gave the LCD screen room to display any images with negligible disruption from EM-Comm coding patterns. In here, we encode a total of 34 bits as shown in panel (d) where the first 2 bits are used for threshold estimation purposes and the remaining 32 bits are an IP address used as an access token to retrieve the corresponding information from the screen. Panel (e) is the EM signal measured by using a EM-Comm wristband corresponding to the image in panel (d). The image in panel (d) is 1920 x 1080 size presented in 1920 x 1200 screen size and this image size makes the first and last 60 rows of pixels on the screen black. The HSYNC is used here to help the receiver wristband detect the boundary and data embedding areas. The envelope of the data embedding area is extracted in panel (e). Then, by combing the envelope of EM signals for the time duration of every 10 pixel rows, the total energy is therefore as depicted in panel (f), where each amplitude represents a bit. The decision line is calculated by using the first two bits' amplitudes. We chose the decision value by averaging the first two amplitudes which

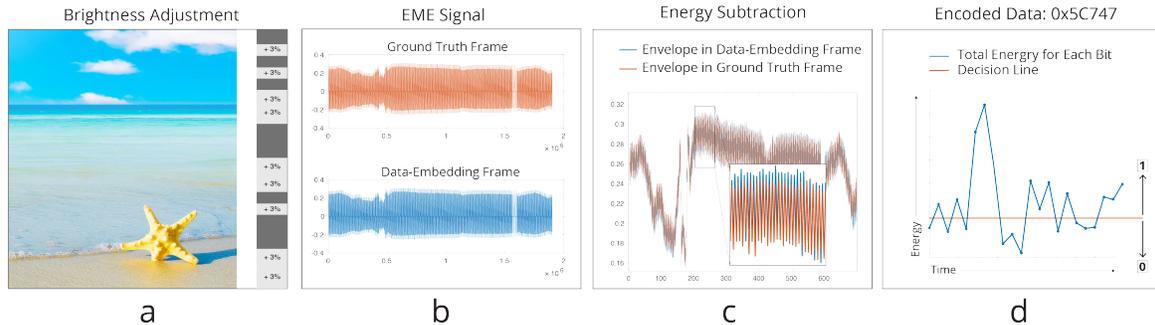


Fig. 8. Overview of the method used to encode EMI data into LCD screens in a way that is “invisible to the eye”. Panel (a) shows how small variations in brightness can be used to encode data into the image in a way that cannot be detected through visible inspection. Panel (b) shows two sequential video frames one with encoded data and the second unmodified frame used as a reference. Panel (c) shows the envelope of the two signals and the resulting difference in amplitude that encodes data. Panel (d) shows the difference between the data frame and the reference frame along the the logical decision threshold boundary.

represent bit 1 and 0 used for determining the amplitude difference. Then, the later amplitudes above or below this decision line can be decoded as bit 1 and 0, respectively.

One should notice that, Figure 7c preserves the amplitude difference made by using black and white bars through X310 measurement with 100 MSPS sampling rate while the Figure 7e has distortion due to using a slow ADC at 2.3 MSPS, as EM-Comm wristband is equipped with. Although a 2.3 MSPS ADC is not fast enough to capture all the peaks in each EM pulse, we leveraged the summation of amplitudes to differentiate EM signals such that the EM-Comm wristband is still able to decode visible data information. We adopted the fact that the summation of amplitudes in the white bar zones should be larger than in black zones to assist slow ADC in decoding binary data. For example, all the amplitudes of EM signal in 10 pixels row when using white bars have larger signal strength than in the black zone. The distortion introduced by using a slow ADC yields the same effect in black and white zone, and the distortion degree of amplitude is random.

Hence, if we use more EM signals as well as more rows to represent binary data 1 and 0 (i.e white and black), the amplitude summation of each area can still reflect the energy difference indicating the binary data 1 and 0. Faster ADCs can usually use fewer rows in amplitude modulation. For example, by using 350×2 rows as white and black for binary data coding, the receiver needs to use an ADC with 100 MSPS to see the amplitude difference. By using 350×6 rows as white and black, the receiver only needs an ADC with 10 MSPS. Thus, by using more rows to represent binary data, the ADC criteria can be relaxed. In addition, if we choose a whole row rather than just a portion rows such as 1920×6 as white and black, the 2.3 MSPS ADC is also able to decode the data by checking amplitude difference. This is because the amplitude difference from using whole rows of black and white bars is large enough against the 2.3M slow ADC distortion.

In short, the distortion due to slow ADCs can be alleviated by using more rows or a higher portion of each row such that the total summation of EM amplitudes has a distinguished difference to help the receiver decode data.

6.2 Embedding Invisible Data in LCD Screen

Although we are able to encode data in any image, the previous approaches generates various black and white bars that can be seen by humans. However, we can also embed data by slightly changing the pixels brightness between each frame such that humans cannot see any encoding information. We leverage the concept that the

human eye is not able to perceive small contrast differences. We modify an image by changing the image contrast in the alpha channel. Existing works have shown that by flipping the alpha channel by 3% or less, humans are not able to perceive differences in images nor video [20]. Although these prior works use cameras to decode data, we use the EME property to decode the data. In LCD screens, when displaying the same content, every adjacent frame has the same magnitude but with different polarity in terms of voltage [6]. This is called frame inversion and aims to prevent pixel damage from applying the same voltage over a long time period. The polarity inversion results in every other frame having identical EM signal impulses when the same image is displayed. Therefore, we select every other two frames as pairs to encode data (i.e 1st and 3rd frames or 2nd and 4th frames) where one frame is used as a ground truth image while the other is used to encode data by changing the contrast. For any still image, we use a 1920x54 pixels block to embed each bit with the 3% brightness enhancement of original images. The brightness enhancement is calculated by using the alpha blending approach as shown below.

$$\text{Brightness Enhancement: } p_{new} = \alpha \times 255 + (1 - \alpha) \times p_{old} \quad (2)$$

Where the p_{old} is the original pixel value in terms of R, G, B channel and p_{new} is the new pixel value by increasing α brightness enhancement. For example, if a pixel value in R,G,B as (200,200,200), the $\alpha = 10\%$ result in a new pixel as (205,205,205) which makes the pixel brighter. Then, our encoder varies the brightness based on the binary data. When a bit "0" is present, the image is not changed. When a bit "1" is present, a portion of the image is increased in brightness $\alpha = 3\%$ with bar size 1920x54 pixels. A 1920x1200 screen size can embed 22 bits by using a bar size 1920x54 for each bit. Here, we use Figure 8 as an example illustration. In panel (a), we embed the binary data "0x5C747" with additional two bit "0" and "1" in preposition to the "0x5C747". The first two bits "0" and "1" are used to help the receiver calculate the threshold for decoding the later 20 bits. Once the whole image has embedded the information data by following the binary sequence "0x5C747", our system generates a video by compiling the original images and coding images in the following order: "Original , Original , Coding, Coding..."

6.2.1 Decoding Invisible Data in USRP X310. During display of the video on the LCD screen, the human eye is not able to detect the difference across different images and only sees a still image. When a user is touching the LCD screen displaying the invisible coding images, the receiver records all the incoming EM signals. To differentiate the amplitude difference of EM signals made by equation (2) across different frames, the receiver need to be able to sense each peak amplitude of EM signals. This results in that a fast ADC is needed and the EM-Comm wristband is not able to decode invisible data. Thus, the receiver we used here is Ettus X310 machine equipped with LFRX daughterboard which has 100 MSPS sampling rate to obtain each peak of EM signals. A user wearing USRP-based wristband to extract the invisible data is illustrated in Figure 8b where a user wore a copper tape wristband and the EM-signal is sampled by X310 machine through RF coaxial cable. A desktop is used to process the EM signals collected from X310 machine. In this scenario, the X310 machine is not battery powered hardware and shared the ground with LCD screen through the power line connection.

Since there is one unmodified and modified image in every two images repeatedly, the USRP X310 can align these two images for further decoding purposes as illustrated in Figure 8b where the ground truth frame represents the original image which does not have any modification, while the data-embedding frame changes a portion of the image's brightness based on the data information. Then, by selecting each frame's EM signal envelope and subtracting from each other, the remaining signal preserves the contrast that we change for data embedding. Because each EM signal only has a slight amplitude increment when increasing a small contrast such as $\alpha = 3\%$, accumulating all the increments in amplitude after subtraction based on each bit range can robustly reflect the data embedding. In panel (d), the total energy after subtracting the ground truth for each bit preserves the brightness variation. The binary decision line is calculated by using the first two accumulation amplitudes representing the bit "0" and "1" which we have used to assist in the receiver decoding procedure. By choosing the

(a) LCD Visible Data Encoding

Category	BPF	WH	NWH
350x2	600	51.2%	49.3%
1920x2	600	2.4 %	3.3 %
350x6	200	13.6%	12.8%
1920x6	200	0 %	0 %
350x10	120	1%	1.4%
1920x10	120	0 %	0 %

(b) LCD Invisible Data Encoding

Category	Alpha	BPF	WH	NWH
1920x10	10	120	0.3%	2.5%
1920x30	10	40	0%	0.5%
1920x54	10	22	0 %	0 %
1920x10	3	120	41.2%	42.5%
1920x30	3	40	14.5%	12.3%
1920x54	3	22	1.2 %	3.3 %

Table 2. LCD BER performance

middle of the first two amplitudes as a threshold, the amplitude below the threshold will be viewed as bit "0" while the amplitude beyond the threshold will be viewed as bit "1". The remaining 20 bits are decoded correctly as 0x5C74. In this setup, all the EM signals are collected and processed in a host computer to run the above decoding approach. In the future, it is worth considering the use of a peak detector, which can hold each incoming arrival peak such that the ADC requirement can be reduced down to 2.3 MSPS and the invisible information can be decoded by using a stand-alone wristband.

6.3 Performance

In this section, we tested the LCD primitive performance through the BER measurement. The procedure is the same as previous primitives. We let a user to randomly touch the LCD screen four times for each different encoding setup as well as hand. Each trial had at least 1 seconds touch to have 60 frames for data collection. The BER is calculated by combining four trials and each trial extract 250 bits to have total 1000 bits for error examination. The visible coding performance is tested by using stand-alone EM-Comm wristband while the invisible coding performance is tested by using USRP X310.

6.3.1 Visible Coding Performance. In this section, we evaluate our system performance based on different pixel configurations as well as data rates. A user wearing an EM-Comm wristband in his/her left hand uses different hands to test the system performance in terms of bit error rate. Namely, the wristband hand here is the left hand while the non-wristband hand is the right hand. In addition, we denoted BPF as bit per frame to describe how many bits can be transmitted for each coding frame by adopting 1920x1200 LCD screen size. For visible coding communication, we test the system performance based on different portions of pixels for embedding data as illustrated in Table 2a. Each frame is fully encoded by setting binary data stream 1 and 0 in turns which produced three different data amount as 600, 200 and 120 bits. The overall BER regarding wristband hand or non-wristband hand both have similar performance, because the signal strength emitted from the LCD screen is able to pass through the whole body. When 350x2 pixels are used to embed binary data 1 and 0, the amplitude difference of EM signals didn't reflect the data information. This is due to the distortion from using a 2.3 MSPS ADC in our EM-Comm wristband and the amplitude difference that randomly appeared. Thus, the bit error rate is up to 50 % which is equal to randomly guessing the binary data. When using a larger portion of pixels in each row such as 1920x2, the amplitude difference made by binary data is detectable against the amplitude distortion from using a slow ADC and the bit error rate is decreased down to around 2 %. Using 1920x2 pixels for embedding binary data has much better performance compared to 350x2, but using 1920x2 pixels for embedding each bit does not permit sharing the screen with other images. However, by increasing the portion of pixels in each column such as using 350x6 and 350x10, the screen can use the remaining portions for displaying any desired images like the example shown in Figure 4d. However, the downside is that the bits per frame is reduced to 200 bits and 120 bits

respectively. One should notice that when more pixels are used for each bit, the result is that the bit error rate can be improved but the the amount of data that can be embedded is reduced and the available space to put any regular image is also reduced.

From experiment results, using 350x10 pixels to embed binary data is recommend to be used, since it can give low BER while only occupying small portion of LCD display. Given a 34 bits packet size for each frame as illustrated in Figure 6a, the packet loss rate is around 28% when using wristband hand to touch LCD screen. However, since typical LCD has 60 Hz refresh rate or higher, a second long touch can ensure a user having successfully information retrieval such as IP address even with the 28 % packet-loss rate. In the future, we will adopt a peak detector to relax ADC criteria such that we can use a very small portion of the screen to embed data while having low BER as well as packet-loss rate.

6.3.2 Invisible Coding Performance. We tested the invisible coding performance with different pixel configurations. The measurement is made by a user wearing a copper-based wristband that is connected to the USRP X310 with an RF cable. The overall performance is shown in Table 2b. When we increase the brightness by choosing $\alpha = 10\%$, the remaining amplitude after subtracting the ground truth image is much larger than $\alpha = 3\%$ such that the bit error rate is also improved. However, the brightness increment yields a noticeable difference to the eye between the normal and modified image. When choosing to increase the brightness using $\alpha = 3\%$, the remaining amplitude of EM signals after subtraction is very small compared to the $\alpha = 10\%$. Thus, when using 1920x10 with $\alpha = 3\%$ brightness enhancement, the BER for each hand has more than 40% error. This is because the amplitude difference after subtracting the ground truth image is too small to preserve the data information. When we choose more pixels for each bit such as 1920x30, the bit error rate can be reduced down to 14.5% in wristband hand. From our experimental results, we found that using 1920x54 pixels to embed binary data '1' and '0' can yield lowest BER. Given a 22 bits packet size for each frame as illustrated in Figure 6b, the packet loss rate is around 24% when using wristband hand. With LCD has 60 Hz refresh rate or higher, a second long touch can ensure a user having successfully information retrieval for small data information. In the future, we will adopt a peak detector to relax ADC criteria and more advanced RF sensing design such that a stand-alone EM-Comm wristband can be used to decode invisible EME data.

7 PRIMITIVES-III: MOTORS AND POWER SUPPLIES

Here, now we consider motors and power supplies as primitives for the EM-Comm system, which can both embed data by switching between different power levels. A user can touch a power adapter or stepper motor to receive the information as illustrated in Figure 4e and f.

7.1 Motors

Motors are one of the oldest forms of electronic devices and importantly provide large EM signals that can be received over longer distances. Common devices containing motors include power tools and many kinds of transportation vehicles. At the most basic level, motors consist of inductive coils which produce large magnetic fields. While it is possible to modulate the rate of rotation with brushed DC motors (commonly found in drills), the data rate is rather low due to the mechanical inertia of the motor. A faster method is to energize and de-energize the coil windings quickly to produce EMI. This is easily done with stepper motors since the drivers are already capable of producing pulses to turn on and off coils. In addition, simply energizing and de-energizing the coils produces no movement nor sounds, but instead only induces a large EMI impulse for embedding data. This on/off coding mechanism can be used in all types of motors.

Generally, an energizing and de-energizing encoding approach uses on-off power for encoding. As an example, we use a RepRap 3D printer (SAV MKL), which use a stepper motor. Then, we use the G-code command G17 (Coil is on) and G87 (Coil is off) to generate two different states. When the coil is charged, the stepper motor randomly

emits EM impulse signals such that it increases the noise floor as illustrated in Figure 9 left panel. When the coil is not energized, the EM signals are not emitted and the noise floor is thus low. We define binary bit data as "1" and "0" when turning the coil on and off respectively.

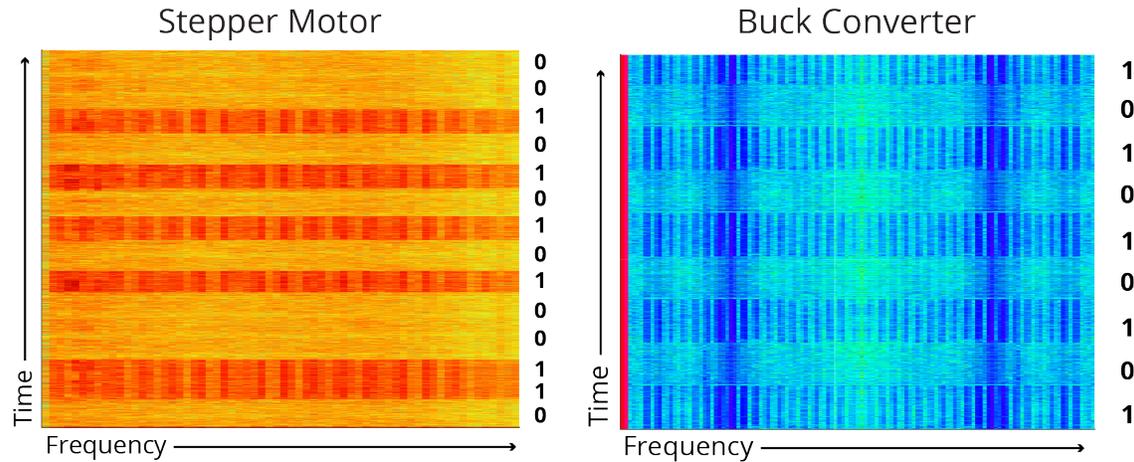


Fig. 9. The left image shows a water fall plot of a stepper motor transmitting data via modulated EMI while the right panels shows a water fall plot of a switch mode power supply (i.e. power brick) transmitting data via its EMI.

The receiver used an FFT window to calculate the noise floor. When the noise floor is high, the receiver decodes the data as "1". Similarly, when the noise floor is low, the receiver decodes the data as "0". In here, our FFT window size is 1024 and our results showing the data embedding is presented in Figure 9 Stepper Motor as an FFT waterfall plot. The highest data rate we can achieve by using RepRap step motor is around 50 Hz.

7.2 Power Supply

Switching-mode power supplies are one of the basic building blocks of electronic devices. They efficiently regulate power from one voltage level to another by dynamically switching current through an inductor, which produces large magnetic fields. This data can travel long distances through the power lines present in homes. An example of this type of signal can be seen in the the waterfall plot of Figure 9 right panel. Thus we can modulate between high and low current loads with different frequencies to embed data 1 and 0, regardless of the device. Here, we chose a regular power supply which takes an AC voltage of 100-240V as input and generates a 12V DC output voltage with maximal current of 1.7 A. Then, we used an HP 6063B DC electronic load machine to act as a host machine such as a computer, which can consume different currents as well as powers. Here, we programmed the host machine consume a 0.1 A and 0.35 A current load repeatedly with 100 Hz switching rate. When the host machine consumes 0.35 A, the EM signals emitted from the power supply have multiple frequency harmonic components. When the host machine consumes a current of 0.1 A, the EM signals don't noticeably have harmonic components and have smaller magnitudes in frequency distribution. Thus, we can embed binary data "0" and "1" by using 0.1 A and 0.35 A respectively. Alternatively, one can also varies two different computation load in computers to enable power supply consuming different currents such that a binary data can be embedded. The EM-Comm wristband decodes the EM data by checking whether the harmonic frequency distribution is in

present as illustrated in Figure 9 Buck Converter with 1024 FFT window size. Given this approach, binary data can therefore be decoded correctly.

7.3 Performance

We check the system performance by testing the stepper motor and power supply when using different hands to perform the interaction. The evaluation is measured by letting a user randomly touch each object for four rounds. Each round made a physical contact long enough to receive more than 250 bits. The combination of each trial gave 1000 bits for BER calculation. Since the EM signals emitted from the power supply and stepper motor are much stronger than emitted from the micro controller I/O line, an FFT window size of 1024 is able to decode data without error as shown in the Table 3. Given a 20 bits packet size, the packet loss rate is 0 % for both wristband hand and non-wristband hand to touch the power supply and stepper motor.

Table 3. Power Supply and Buck Converter

Category	Data Rate (bps)	WH	NWH
Stepper Motor	50	0%	0%
Buck Converter	100	0%	0%

8 APPLICATIONS

To demonstrate the utility of the EM-Comm system we developed a number of applications intended to explore the types of interaction modality enabled by this technology. While these usage scenarios are meant to be illustrative the underlining technology was developed into fully working demos as shown in figure 10 and the video figure accompanying this article.

In the first scenario visitors looking for a particular building on a large campus are consulting a public display consisting of a LCD screen. Once they have selected the building of interest the EM-Comm icon is displayed and the user touches the screen to capture the EMI data encoded into the image. The EM-Comm wrist band launches the maps application he user phone and they is directed to the correct building.

Once the visitor arrive he needs to print a document from his phone but since he is new to the office he has not connected to the printer before. By touching the print button on the printer EMI data is transmitted from the printer through his body and into the wristband which in turn connects his mobile phone to the printer in order to send the document. Unfortunately the printer jams and the alert LED blinks indicating a problem. In order to gain more information the user touches the LED which transmit detailed diagnostic information via EMI modulation. A tutorial is then launched on the users phone guiding him through the repair process, finally allowing him to print the document.

In the final application the user is unfamiliar with the office coffee machine. In order to gain a better understanding of its operation he lightly touches the button which transmits it's functionality via modulated EMI to the wrist band which displays the function of the button on the user phone in his native language. In the following section we discuss the usage modes that we used to build these example application.

8.1 Pairing

One of the most common, and sometimes tedious, tasks is the pairing of wireless devices. With EM-Comm this process is simplified by leveraging touched based data transfer. For example consider the case when a user wants to print a document from his/her smart phone on an unknown printer. Once the appropriate document is in the foreground on the smart phone, the user simply needs to press the EM-Comm enabled "print" button on the printer. This action precisely indicates which device is of interest (i.e. this printer, out of all printers on



Fig. 10. Images of the three demo applications developed using the EM-Comm technology. They include (a) pairing, trouble shooting, and printing from a Mobile phone, (b) grabbing a map & directions from a public LCD display and sending them to the users smart phone, and (c) launching virtual tool tips of button functionality on a coffee machine

the network), what action should be performed (i.e. printing), and when it should take place. Using the button primitive and spread spectrum frequency shift keying modulation scheme described above, the printer transmits its IP address along with its make and model through users hand and into the EM-Comm wrist band receiver where it is decoded. This information is then passed onto the smartphone which makes a connection and prints the document.

8.2 Authentication

Most wireless radio systems attempt to transmit data as far as possible making them vulnerable to easy dropping. In contrast EM-Comm enabled devices offer an extra layer of privacy by only transmitting data through touch. While a sophisticated attacker could use high-end equipment to “sniff” the EMI data emitted by a given device. Due to practical limitation such as the decay of the EMI signal with distance and background noise this would only be effective up to a few meters, which would require physical access to the targets office or home. When coupled with a layered security strategy the EM-Comm device can transmit a key on its hidden channel and the EM-Comm wrist band can reply via a public channel such as Bluetooth or WiFi. Furthermore, since both the target device and the wrist band can detect when a user presses one of the target device’s buttons, temporal co-occurrence can be used to aid in authentication.

Returning to the previous example when the users goes to press the “Print” button, the printer is continuously transmitting information like its model number, IP address and button press counter. When the button is press the EMI signal is grounded and disappears. The time of this event and approximate duration is detected by both the printer and the EM-Comm wrist band. Once the button is release the user’s finger will easily remain in contact long enough for the printer to transmit its incremented button press counter, along with secret key. These two unique pieces of information can be used to initiate a secure authentication round.

8.3 Enhanced State Indicators

Often times LED indicators are used to provide valuable state information about the device but they are very limited in how expressive they can be (typically only on, off and blinking). While this may be enough if there is proper supporting text written next to the LED. More often than not, users must search internet forms or the manual to decipher their meaning. As sold the printer used in our example application uses the “Power” LED and “Alarm” LED to denote 45 different error codes (i.e. Support Codes) through a sequence of flashes and pauses. By using our custom EM-Comm firmware we are able to transmit diagnostic data through the LED via modulated EMI. When the indicator LED is touched by the user the error codes are sent through the body and decoded by the wrist band and then displayed on a smart phone or computer. Error messages could also take the form of URLs that point to the manufactures website for technical support.

8.4 Device and Stated Specific Applications

Understanding what devices a user is interact with and the stated of those devices, provides a great deal of contextual information about what the user intentions. This offers the opportunity to launch device and task specific applications. For instance in the public display scenario when the user touches the screen and captures the EMI data embedded in the video. The EM-Comm wrist band decodes the data and send a command to the user's smart phone which launches the maps application. The application is automatically populated with the location shown on the public display and the phone start directing the user to his/her destination.

8.5 Device Selection

An extension of device pairing is device selection through physical interaction. In this scenario a group of people are deciding where to go for dinner. One person wants to send info on a restaurant from her phone to a subset of the group of friends standing in front of her. Here she touches her smart phone which is modulating the EMI from its LCD screen to encode a pointer to the selected content. She then specifies which of her friends should get the directions by touching there smart phones. In the scenario the friend's phone would be modulating its screen to encode its IP address in its EMI. This data would be received on physical contact and location data would be transmitted via the internet.

8.6 EMI Steganography

Steganography is the practice of hiding data inside other forms of data. One typical example is hiding text in an image file by encoding the text in the least significant bits of the pixel's RGB values. If done correctly this manipulation is imperceptible by people. As described in section 6.2 it is possible to embed data in a video such that it will manipulate the row and column drivers of the display to emit modulated EMI data. This would allow digital content to be "water marked" such that it a user to retrieve data when touching the screen such a movie metadata or pointers to web based content.

8.7 Tool Tips & Tutorial

Despite the best efforts of the manufactures of appliance and devices, the symbols used for buttons can be confusing and the text may not be in the language of user understands. To explore this scenario we augmented an office coffee machine to EMI encoded data through its capacitive touch buttons. Here when a user hovers over the button the coffee machine transmits the function (e.g. tool tip) of the button via modulated EMI, and the EM-Comm wristband sends the translated text to the user's smart phone. This tool tip functionality can be applied to any situation where limited user interface needs to be augmented with extra information. Likewise, EMI data transmitted through the buttons could be used in concert with a mobile phone, to help guide a user through a tutorial on how to use the an appliance or device.

9 CONCLUSION

The way we manipulate and physically interact with our everyday electronic devices encodes a great deal of contextual information about our intentions. For example pressing the "pairing button" on a Bluetooth peripheral precisely indicates which device is of interest, what action should be performed, and when it should take place. Unfortunately our mobile computing devices, such as smart phones and smart watches, have very little ability to detect and thus react to these touch interaction events. While systems using RFID tags or body channel communication methods can provide touch-based communication they all require the object and/or device of interest to incorporate dedicated radio transmitters or be manually tagged.

In contrast the EM-Comm system simplifies the deployment scenario by only requiring the user be instrumented with a smart watch form factor receiver. Target devices need no modification or radio hardware of any type.

However, they do need to have one of the six basic EMI primitives (i.e. I/O lines, an LED, a button, LCD screens, motors, or a switch mode power supply) whose operation can be modified in software. To enable EM-Comm touch-based wireless communication the target device only needs a simple software modification where data is encoded into the devices preexisting EMI using our spread spectrum frequency shift keying approach. With the exception of the LCD screen which only needs to play a video or image with EMI data encoded as either visible or invisible changes in pixel row brightness.

Furthermore, the EM-Comm receiver hardware only consists of electrodes for coupling to the skin, a low cost operational amplifier for signal gain and a processor with a low end analog to digital converter, which makes integration in to a production smartwatch or wearable device a straight forward process. Ultimately, EM-Comm enables nearly any electronic device to be turned into a touch-based radio transmitter with only a software upgrade.

REFERENCES

- [1] 2017. *FCC's Title 47 of the Code of Federal Regulations, Part 15*. Technical Report. Federal Communications Commission.
- [2] Michael Buettner, Richa Prasad, Matthai Philipose, and David Wetherall. 2009. Recognizing Daily Activities with RFID-based Sensors. In *Proceedings of the 11th International Conference on Ubiquitous Computing (UbiComp '09)*. ACM, New York, NY, USA, 51–60.
- [3] Ke-Yu Chen, Gabe A. Cohn, Sidhant Gupta, and Shwetak N. Patel. 2013. uTouch: Sensing Touch Gestures on Unmodified LCDs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2581–2584. <https://doi.org/10.1145/2470654.2481356>
- [4] Gabe Cohn, Daniel Morris, Shwetak Patel, and Desney Tan. 2012. Humantenna: Using the Body As an Antenna for Real-time Whole-body Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 1901–1910. <https://doi.org/10.1145/2207676.2208330>
- [5] Gabe Cohn, Daniel Morris, Shwetak N. Patel, and Desney S. Tan. 2011. Your Noise is My Command: Sensing Gestures Using the Body As an Antenna. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 791–800. <https://doi.org/10.1145/1978942.1979058>
- [6] David J. R. Cristaldi, Salvatore Pennisi, and Francesco Pulvirenti. 2009. *Liquid Crystal Display Drivers: Techniques and Circuits* (1st ed.). Springer Publishing Company, Incorporated.
- [7] Paul Dietz and Darren Leigh. 2001. DiamondTouch: A Multi-user Touch Technology. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology (UIST '01)*. ACM, New York, NY, USA, 219–226. <https://doi.org/10.1145/502348.502389>
- [8] Equipment Authorization Division. 1996. *Understanding the FCC Regulations for Low-Power Non-Licensed Transmitters*. OET Bulletin NO. 63. Federal Communications Commission, Office of Engineering and Technology; Customer Service Branch, MS 1300F2; 7435 Oakland Mills Road; Columbia, MD 21046.
- [9] Kenneth P. Fishkin, Matthai Philipose, and Adam Rea. 2005. Hands-On RFID: Wireless Wearables for Detecting Use of Objects. In *Proceedings of the Ninth IEEE International Symposium on Wearable Computers (ISWC '05)*. IEEE Computer Society, Washington, DC, USA, 38–43.
- [10] James Fogarty, Carolyn Au, and Scott E. Hudson. 2006. Sensing from the Basement: A Feasibility Study of Unobtrusive and Low-cost Home Activity Recognition. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology (UIST '06)*. ACM, New York, NY, USA, 91–100. <https://doi.org/10.1145/1166253.1166269>
- [11] Tobias Grosse-Puppenthal, Sebastian Herber, Raphael Wimmer, Frank Englert, Sebastian Beck, Julian von Wilmsdorff, Reiner Wichert, and Arjan Kuijper. 2014. Capacitive Near-field Communication for Ubiquitous Interaction and Perception. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '14)*. ACM, New York, NY, USA, 231–242. <https://doi.org/10.1145/2632048.2632053>
- [12] Tobias Grosse-Puppenthal, Christian Holz, Gabe Cohn, Raphael Wimmer, Oskar Bechtold, Steve Hodges, Matthew S. Reynolds, and Joshua R. Smith. 2017. Finding Common Ground: A Survey of Capacitive Sensing in Human-Computer Interaction. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 3293–3315. <https://doi.org/10.1145/3025453.3025808>
- [13] Sidhant Gupta, Ke-Yu Chen, Matthew S. Reynolds, and Shwetak N. Patel. 2011. LightWave: Using Compact Fluorescent Lights As Sensors. In *Proceedings of the 13th International Conference on Ubiquitous Computing (UbiComp '11)*. ACM, New York, NY, USA, 65–74. <https://doi.org/10.1145/2030112.2030122>
- [14] Sidhant Gupta, Matthew S. Reynolds, and Shwetak N. Patel. 2010. ElectriSense: Single-point Sensing Using EMI for Electrical Event Detection and Classification in the Home. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing (UbiComp '10)*. ACM, New York, NY, USA, 139–148. <https://doi.org/10.1145/1864349.1864375>

- [15] Simon Haykin. 2009. *Communication Systems* (5th ed.). Wiley Publishing.
- [16] Mehrdad Hesar, Vikram Iyer, and Shyamnath Gollakota. 2016. Enabling On-body Transmissions with Commodity Devices. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '16)*. ACM, New York, NY, USA, 1100–1111. <https://doi.org/10.1145/2971648.2971682>
- [17] Christian Holz and Marius Knaust. 2015. Biometric Touch Sensing: Seamlessly Augmenting Each Touch with Continuous Authentication. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 303–312. <https://doi.org/10.1145/2807442.2807458>
- [18] Gierad Laput, Chouchang Yang, Robert Xiao, Alanson Sample, and Chris Harrison. 2015. EM-Sense: Touch Recognition of Uninstrumented, Electrical and Electromechanical Objects. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 157–166. <https://doi.org/10.1145/2807442.2807481>
- [19] Hanchuan Li, Eric Brockmeyer, Elizabeth J. Carter, Josh Fromm, Scott E. Hudson, Shwetak N. Patel, and Alanson Sample. 2016. PaperID: A Technique for Drawing Functional Battery-Free Wireless Interfaces on Paper. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 5885–5896. <https://doi.org/10.1145/2858036.2858249>
- [20] Tianxing Li, Chuankai An, Xinran Xiao, Andrew T. Campbell, and Xia Zhou. 2015. Real-Time Screen-Camera Communication Behind Any Scene. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '15)*. ACM, New York, NY, USA, 197–211. <https://doi.org/10.1145/2742647.2742667>
- [21] Nicolai Marquardt, Alex S. Taylor, Nicolas Villar, and Saul Greenberg. 2010. Rethinking RFID: Awareness and Control for Interaction with RFID Systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 2307–2316. <https://doi.org/10.1145/1753326.1753674>
- [22] N. Matsushita, S. Tajima, Y. Ayatsuka, and J. Rekimoto. 2000. Wearable key: device for personalizing nearby environment. In *Digest of Papers. Fourth International Symposium on Wearable Computers*. 119–126. <https://doi.org/10.1109/ISWC.2000.888473>
- [23] Duck Gun Park, Jin Kyung Kim, Jin Bong Sung, Jung Hwan Hwang, Chang Hee Hyung, and Sung Weon Kang. 2006. TAP: Touch-and-play. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*. ACM, New York, NY, USA, 677–680. <https://doi.org/10.1145/1124772.1124873>
- [24] S. Parlak and I. Marsic. 2013. Detecting Object Motion Using Passive RFID: A Trauma Resuscitation Case Study. *IEEE Transactions on Instrumentation and Measurement* 62, 9 (Sept 2013), 2430–2437. <https://doi.org/10.1109/TIM.2013.2258772>
- [25] Kurt Partridge, Bradley Dahlquist, Alireza Veisesh, Annie Cain, Ann Foreman, Joseph Goldberg, and Gaetano Borriello. 2001. Empirical Measurements of Intrabody Communication Performance Under Varied Physical Configurations. In *Proc. of the 14th Annual ACM Symposium on User Interface Software and Technology (UIST '01)*. 183–190. <https://doi.org/10.1145/502348.502381>
- [26] Shwetak N. Patel, Thomas Robertson, Julie A. Kientz, Matthew S. Reynolds, and Gregory D. Abowd. 2007. At the Flick of a Switch: Detecting and Classifying Unique Electrical Events on the Residential Power Line. In *Proceedings of the 9th International Conference on Ubiquitous Computing (UbiComp '07)*. Springer-Verlag, Berlin, Heidelberg, 271–288. <http://dl.acm.org/citation.cfm?id=1771592.1771608>
- [27] D. J. Patterson, D. Fox, H. Kautz, and M. Philipose. 2005. Fine-grained activity recognition by aggregating abstract object usage. In *Ninth IEEE International Symposium on Wearable Computers (ISWC'05)*. 44–51. <https://doi.org/10.1109/ISWC.2005.22>
- [28] Jun Rekimoto. 1997. Pick-and-drop: A Direct Manipulation Technique for Multiple Computer Environments. In *Proc. of the 10th ACM Symposium on User Interface Software and Technology (UIST '97)*. ACM, New York, NY, USA, 31–39. <https://doi.org/10.1145/263407.263505>
- [29] David Schwarz, Max Schwarz, Jörg Stückler, and Sven Behnke. 2015. *Cosero, Find My Keys! Object Localization and Retrieval Using Bluetooth Low Energy Tags*. Springer International Publishing, Cham, 195–206. https://doi.org/10.1007/978-3-319-18615-3_16
- [30] Andrew Spielberg, Alanson Sample, Scott E. Hudson, Jennifer Mankoff, and James McCann. 2016. RapID: A Framework for Fabricating Low-Latency Interactive Objects with RFID Tags. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 5897–5908. <https://doi.org/10.1145/2858036.2858243>
- [31] Emmanuel Munguia Tapia, Stephen S. Intille, and Kent Larson. 2004. *Activity Recognition in the Home Using Simple and Ubiquitous Sensors*. Springer Berlin Heidelberg, Berlin, Heidelberg, 158–175. https://doi.org/10.1007/978-3-540-24646-6_10
- [32] Edward J. Wang, Tien-Jui Lee, Alex Mariakakis, Mayank Goel, Sidhant Gupta, and Shwetak N. Patel. 2015. MagnifiSense: Inferring Device Interaction Using Wrist-worn Passive Magneto-inductive Sensors. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '15)*. ACM, New York, NY, USA, 15–26. <https://doi.org/10.1145/2750858.2804271>
- [33] C. Yang and A. P. Sample. 2016. EM-ID: Tag-less identification of electrical devices via electromagnetic emissions. In *2016 IEEE International Conference on RFID (RFID)*. 1–8. <https://doi.org/10.1109/RFID.2016.7488014>
- [34] T. G. Zimmerman. 1996. Personal Area Networks: Near-field intrabody communication. *IBM Systems Journal* 35, 3.4 (1996), 609–617. <https://doi.org/10.1147/sj.353.0609>
- [35] Thomas G. Zimmerman, Joshua R. Smith, Joseph A. Paradiso, David Allport, and Neil Gershenfeld. 1995. Applying Electric Field Sensing to Human-computer Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '95)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 280–287. <https://doi.org/10.1145/223904.223940>

Received May 2017; revised July 2017; accepted August 2017