EXPOSYS DATA LABS

DOMAIN- IOT (INTERNET OF THINGS)

TASK- Encryption and Decryption between users in Morse using a Flashlight.

GROUP MEMBERS:-

SHIRISHA.B

R CHETANA NATH

ABSTRACT

The expression “Internet of Things” (IoT), coined back in 1999 by Kevin Ashton, the British technology pioneer who cofounded the Auto-ID Center at the Massachusetts Institute of Technology, is becoming more and more mainstream. In opening the IoT Week 2013 with a pre-recorded video message, Ashton insisted on the realization that IoT is here now; it is not the future but the present. While Gartner identifies IoT as one of the top ten strategic technology trends, Cisco forecasts 50 billion devices connected by 2020, a potential market in excess of $14 trillion, and also claims that IoT is actually already here. Similarly, it is not only companies with a technological focus, such as Ericsson, Bosch or Siemens that use IoT to advertise their cutting edge technologies – media companies such as the BBC are conducting research activities and have plans for IoT deployment. In short, we are currently on the verge of witnessing the emergence of a “mega-market”, where markets such as home and building automation, electricity generation and distribution, logistics, automotive, as well as telecommunications and information technology will steadily converge. As yet, we do not know the consequences of connecting all of these smart objects (smart meter, e-vehicle, cargo container, fridge etc.) to the Internet.

Morse code is a method of transmitting text information as a series of on-off tones, clicks that can be directly understood by a skilled listener or observer without special equipment. It is named for Samuel F. B. Morse, an inventor of the telegraph. Morse code has been employed as an assistive technology, helping people with a variety of disabilities to communicate. Morse can be sent by persons with severe motion disabilities, as long as they have some minimal motor control. An original solution to the problem that caretakers have to learn to decode has been an electronic typewriter with the codes written on the keys. Morse code can also be translated by computer and used in a speaking communication aid. An important advantage of Morse code over row column scanning is that once learned, it does not require looking at a display. Also, it appears faster than scanning.

Table of Contents

Introduction

The Internet of Things (IoT) has become a common news item and marketing trend. Beyond the hype, IoT has emerged as an important technology with applications in many fields. IoT has roots in several earlier technologies: pervasive information systems, sensor networks, and embedded computing. The term IoT system more accurately describes the use of this technology than does Internet of Things. Most IoT devices are connected together to form purpose-specific systems; they are less frequently used as general-access devices on a worldwide network. IoT moves beyond pervasive computing and information systems, which concentrated on data. Smart refrigerators are one example of pervasive computing devices. Several products included built-in PCs and allowed users to enter information about the contents of their refrigerator for menu planning. Conceptual devices would automatically scan the refrigerator contents to take care of data entry. The use cases envisioned for these refrigerators are not so far removed from menu planning applications for stand-alone personal computers. Sensor network research spanned a range of configurations. Many of these were designed for data collection at very low data rates. The collected data would then be sent to servers for processing. Traditional sensor network research did not emphasize in-network processing. Embedded computing concentrated on either stand-alone devices or tightly coupled networks such as those used in vehicles. Consumer electronics and cyberphysical systems were two major application domains for embedded computing; both emphasized engineered systems with well-defined goals. Given the wide range of advocates for IoT technology, no single, clear definition of the term has emerged. We can identify several possibilities :Internet-enabled physical devices, although many devices don’t use the Internet Protocol Soft real-time sensor networks. Dynamic and evolving networks of embedded computing devices This book is primarily interested in IoT systems. We use this term to capture two characteristics. First, the system is designed for one or a set of applications, rather than being an agglomeration of Internet-enabled devices. Second, the IoT system takes into account the dynamics of physical systems. An IoT system may consist primarily of sensors; in some cases it may include a significant number of actuators. In both cases, the goal is to process signals and time-series data. Interest in the Internet of Things has been spurred by the availability of microelectromechanical (MEMS) sensors. Integrated accelerometers, gyroscopes, chemical

sensors, and other forms of sensor are now widely available. The low cost and power consumption of these sensors enables new applications well beyond those of traditional laboratory or industrial measurement equipment. These sensor applications push IoT systems toward signal processing. IoT is also enabled by the very low cost of VLSI digital and analog electronics. As we will see, IoT nodes do not rely on state-of-the-art VLSI manufacturing processes. In fact, they are inexpensive because they are able to make use of older manufacturing lines; the lower device counts available in these older technologies are more than sufficient for many IoT systems. IoT systems must consume very little power. Power consumption is a key factor in total cost of ownership for IoT systems. Achieving the necessary power levels requires careful attention to hardware design, software design, and application algorithms. Security and safety are key design and operational requirements for IoT systems. As we have argued elsewhere, safety and security are no longer separable problems. The merger of computational and physical systems requires us to merge the previously separate tasks of safe physical system design and secure computer system design.

Existing Method

- *Morse code is a method of transmitting text information as a series of on-off tones, lights, or clicks that can be directly understood by a skilled listener or observer without special equipment. The International Morse Code encodes the ISO basic Latin alphabet, some extra Latin letters, the Arabic numerals and a small set of punctuation and procedural signals as standardized sequences of short and long signals called "dots" and "dashes",or "dits" and "dahs". Because many non-English natural languages use more than the 26 Roman letters, extensions to the Morse alphabet exist for those languages.*

- *Morse code is most popular among amateur radio operators, although it is no longer required for licensing in most countries, including the US. Pilots and air traffic controllers usually need only a cursory understanding. Aeronautical navigational aids, such as VORs and NDBs, constantly identify in Morse code. Compared to voice, Morse code is less sensitive to poor signal conditions, yet still comprehensible to humans without a decoding device. Morse is therefore a useful alternative to synthesized speech for sending automated data to skilled listeners on voice channels. Many amateur radio repeaters, for example, identify with Morse, even though they are used for voice communications.*

Proposed method with Architecture

Many newcomers seek exemption from ‘tiresome’ Morse test for Short wave operation. This is unfortunate as the Morse code is the key to enter into the world of ham radio with little monetary investment. A novice can assemble a simple Morse code transmitter with lesser technical hurdles than that of a SSB Voice transmitter. Morse code can be learnt easily if we use certain techniques to remember the codes. Learning the Morse code is also a personal venture embarked upon by alone. Almost all the letters/characters and puncuation marks can be arranged in certain groups which can used to show the resemblance between/among the combination of dot and dashes. For example the letter ‘A’ (. _) is the opposite of ‘N’ (_ .) in Morse code. Similarly, the letters A, U, V and the character 4 can be made into a group which shows a definite sequence. Given below is a table of such combinations.

A	._	T	_	N	_.	E	.
U	.._	M	__	D	_..	I	..
V	..._	O	___	B	_...	S	...
4	...._					H	....
						5	.....
W	.__	Y	_.__	K	_._	P	.__.
G	__.	Q	__._	R	._.	X	_.._
L	._..	Z	__..	A	._	D	_..
F	.._.	J	.___	N	_.	U	.._
		C	_._.				
B	_...						

After remembering the Morse code combination for the EISH combination, the following words can be formed to be sent for receiving practice.

Methodology

Encryption

1. In the case of encryption, we extract each character (if not space) from a word one at a time and match it with its corresponding morse code stored in whichever data structure we have chosen(if you are coding in python, dictionaries can turn out to be very useful in this case)
2. Store the morse code in a variable that will contain our encoded string and then we add a space to our string that will contain the result.
3. While encoding in morse code we need to add 1 space between every character and 2 consecutive spaces between every word.
4. If the character is a space then add another space to the variable containing the result. We repeat this process till we traverse the whole string

Decryption

1. In the case of decryption, we start by adding a space at the end of the string to be decoded (this will be explained later).
2. Now we keep extracting characters from the string till we are not getting any space.
3. As soon as we get a space we look up the corresponding English language character to the extracted sequence of characters (or our morse code) and add it to a variable that will store the result.
4. Remember keeping track of the space is the most important part of this decryption process. As soon as we get 2 consecutive spaces we will add another space to our variable containing the decoded string.
5. The last space at the end of the string will help us identify the last sequence of morse code characters (since space acts as a check for extracting characters and start decoding them).

Implementation

A String called letters which gives the valid characters in the MorseCode. This table is a good candidate for an Alphabet object to ease looking up characters.

An array of Strings called codes which give the series of dashes and dots equivalent to each character in the allowed set of characters.

The central strategy while encrypting is simple: for each character in the message, determine the index of that character and output the equivalent dash-dot string. Each dash-dot is separated by 3 spaces and each word is separated by 7 spaces.

```
Message: "ALL YOUR BASE"

A: ".-"   "   "

L: ".-.." "   "

L: ".-.." "   "

_: "       "

Y: "-.--" "   "

O: "---"  "   "

U: "..-"  "   "

R: ".-."  "   "

_: "       "

B: "-..." "   "

A: ".-"   "   "

S: "..."  "   "
```

```
E: "."



Encrypted: .-   .-..   .-..       -.--   ---   ..-   .-.       -...   .-
...   .
```

Decrypting is a matter of reading dots/dashes until a space is encountered. The sequence of dashes/dots is looked up in the table of equivalences and the equivalent character is added to the growing message. After this, count how many spaces there are prior to the next dot/dash: 3 spaces means the next character is part of the same word, 7 means the next character starts a new word and a space should be put into the growing decoded message.

```
Encrypted: .-   .-..   .-..       -.--   ---   ..-   .-.       -...   .-
...   .

position:  ^

code: ""

mesg: ""



Encrypted: .-   .-..   .-..       -.--   ---   ..-   .-.       -...   .-
...   .

position:    ^

code: ".-" -> A

mesg: "A"



Encrypted: .-   .-..   .-..       -.--   ---   ..-   .-.       -...   .-
...   .

position:       ^

code: "" -> 3 spaces, same word

mesg: "A"
```

```
Encrypted: .-   .-..   .-..       -.--   ---   ..-   .-.       -...   .-
...   .

position:           ^

code: ".-.." -> L

mesg: "AL"



Encrypted: .-   .-..   .-..       -.--   ---   ..-   .-.       -...   .-
...   .

position:              ^

code: "" -> 3 spaces, same word

mesg: "AL"



Encrypted: .-   .-..   .-..       -.--   ---   ..-   .-.       -...   .-
...   .

position:                  ^

code: ".-.." -> L

mesg: "ALL"



Encrypted: .-   .-..   .-..       -.--   ---   ..-   .-.       -...   .-
...   .

position:                        ^

code: "" -> 7 spaces, new word

mesg: "ALL "



Encrypted: .-   .-..   .-..       -.--   ---   ..-   .-.       -...   .-
...   .

position:                            ^
```

```
code: "-.--" -> Y

mesg: "ALL Y"

...
```

The code to accomplish decryption is more intricate than other processes we have previously dealt with. It may be helpful to establish private helper methods to perform tasks such asScan through spaces in a string until the next non-space character and return the number of spaces.

Scan all non-space characters from a given start position and return a string of those non-space characters.While not strictly necessary, such little methods can simplify your the logic of your decryption routine to make it "prettier".

Conclusion

We have attempted to design and implement the task on“Encryption and Decryption between users in Morse using a Flashlight”. python supports enormous flexibility in the design and the use of IOT programs. The presence of many built in classes methods take care of much functionality and reduce the job of coding as well as makes the implementation simpler. The project was started with the designing phase in which we figured the requirements needed, the layout design, then comes the detail designing of each function after which, was the testing and debugging stage. We have tried to implement the project making it as user-friendly and error free as possible.