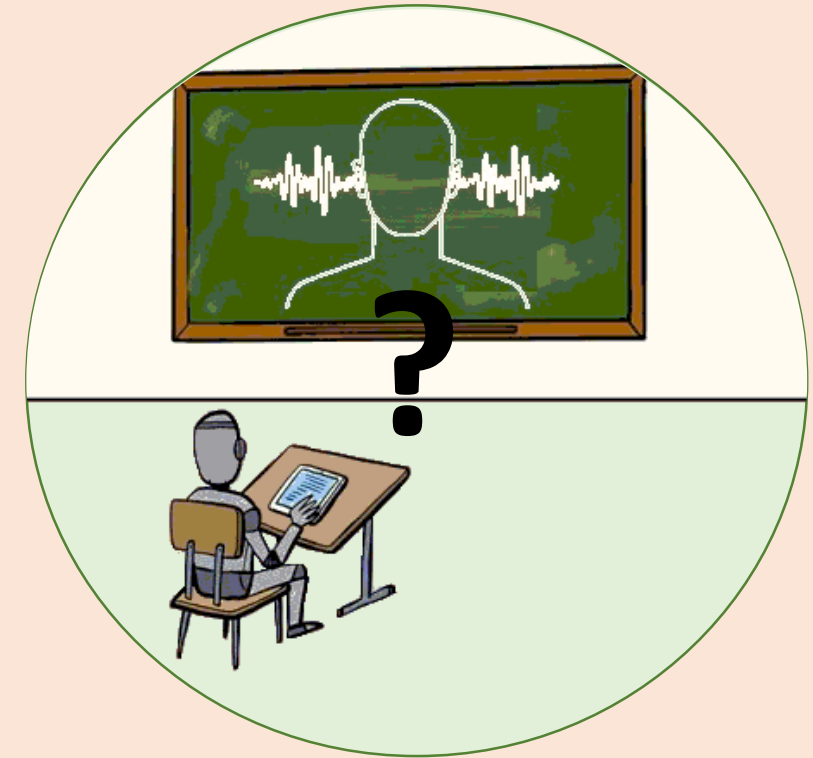


# TEACHING ROBOTS TO HEAR US









ROBOT  
РОБОТ





ROBOT  
РОБОТ





ROBOT  
РОБОТ

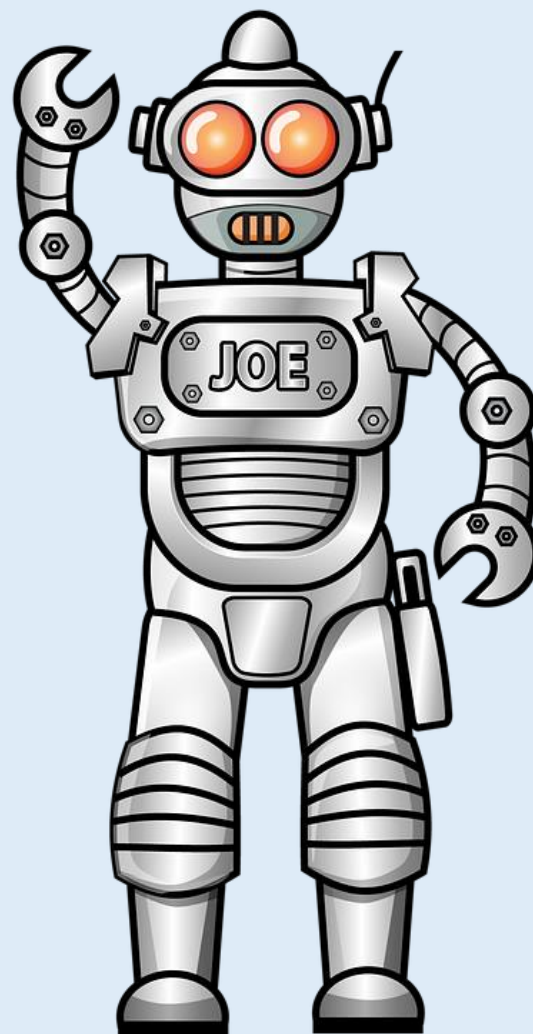
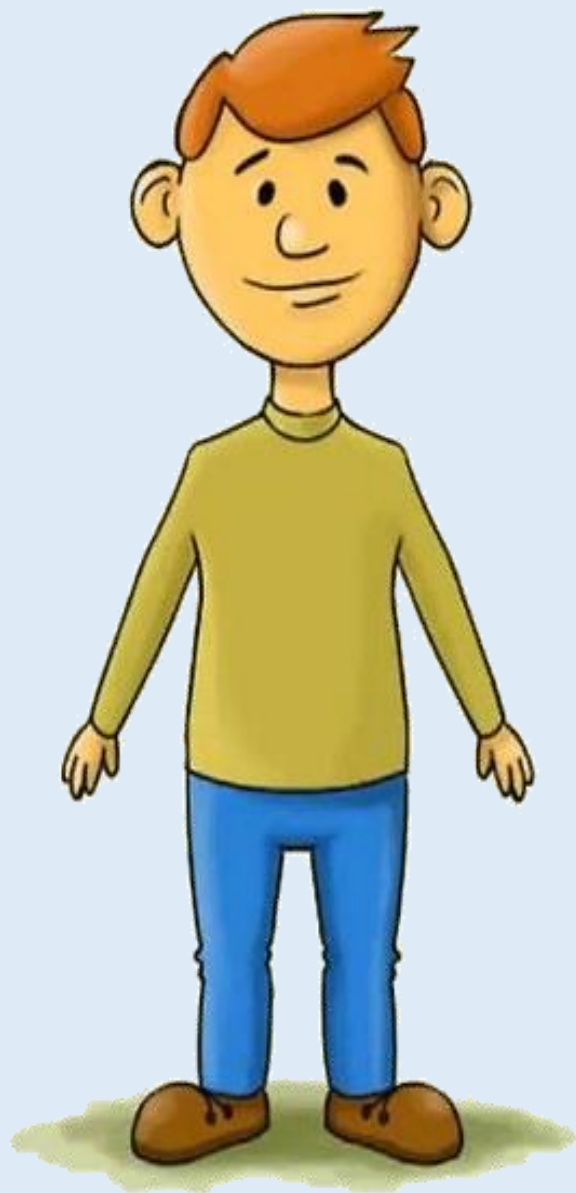


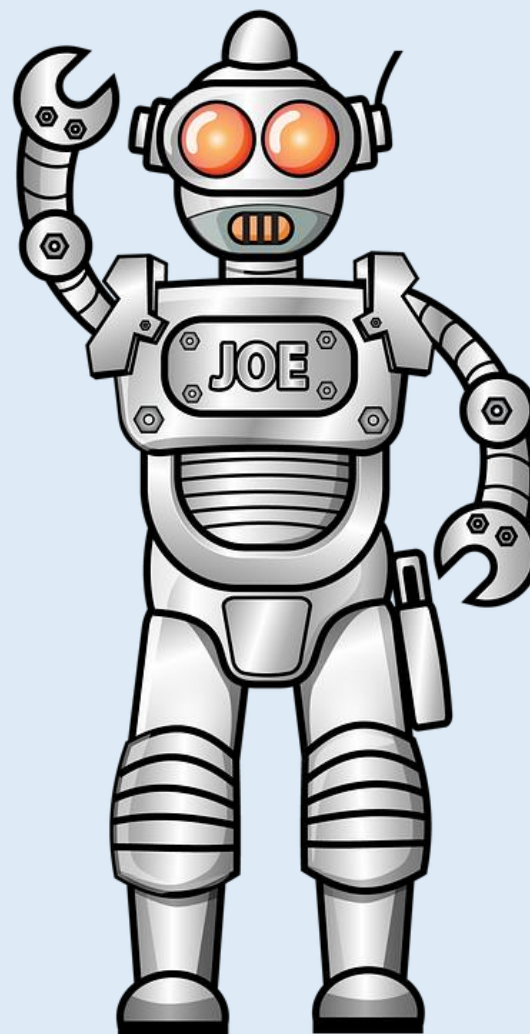
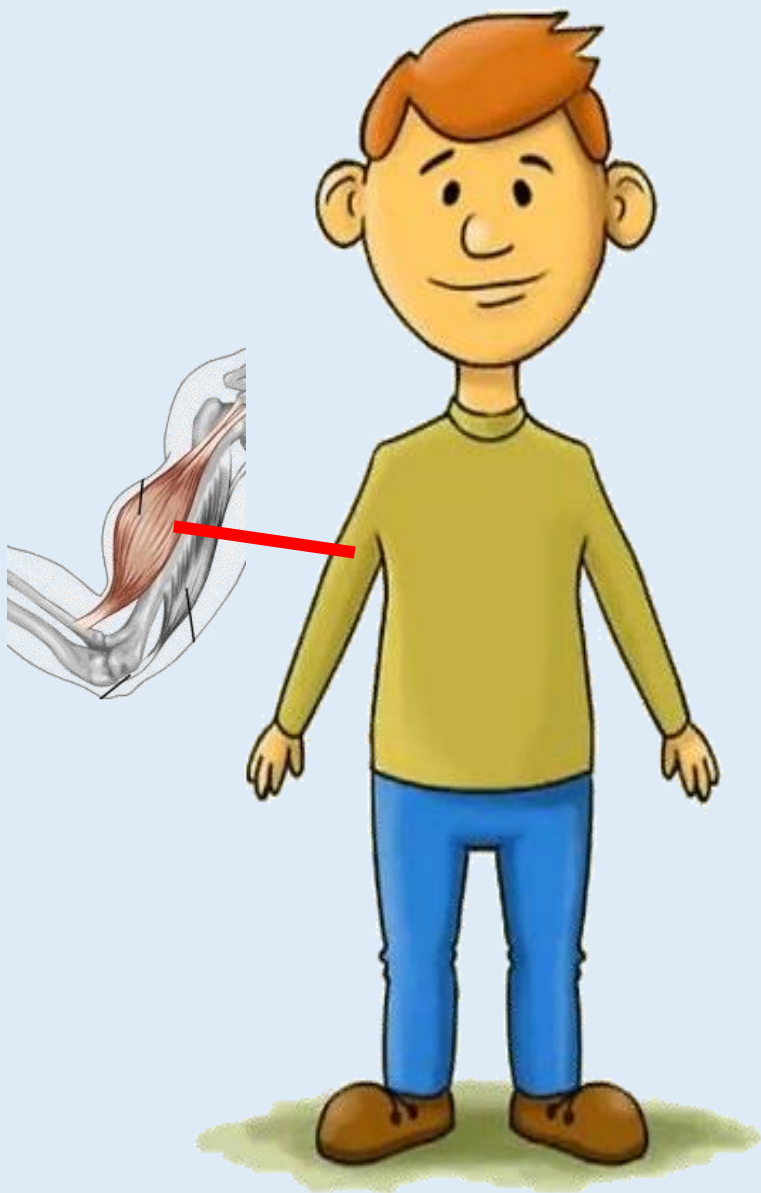


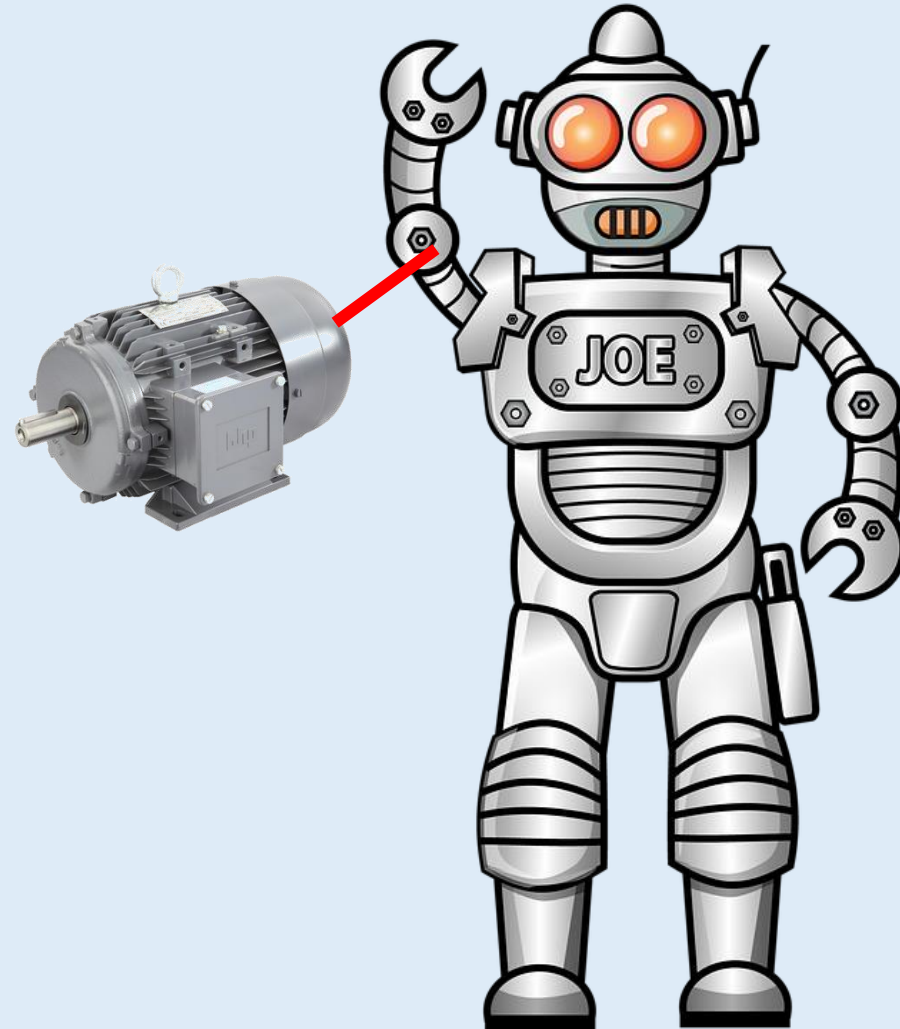
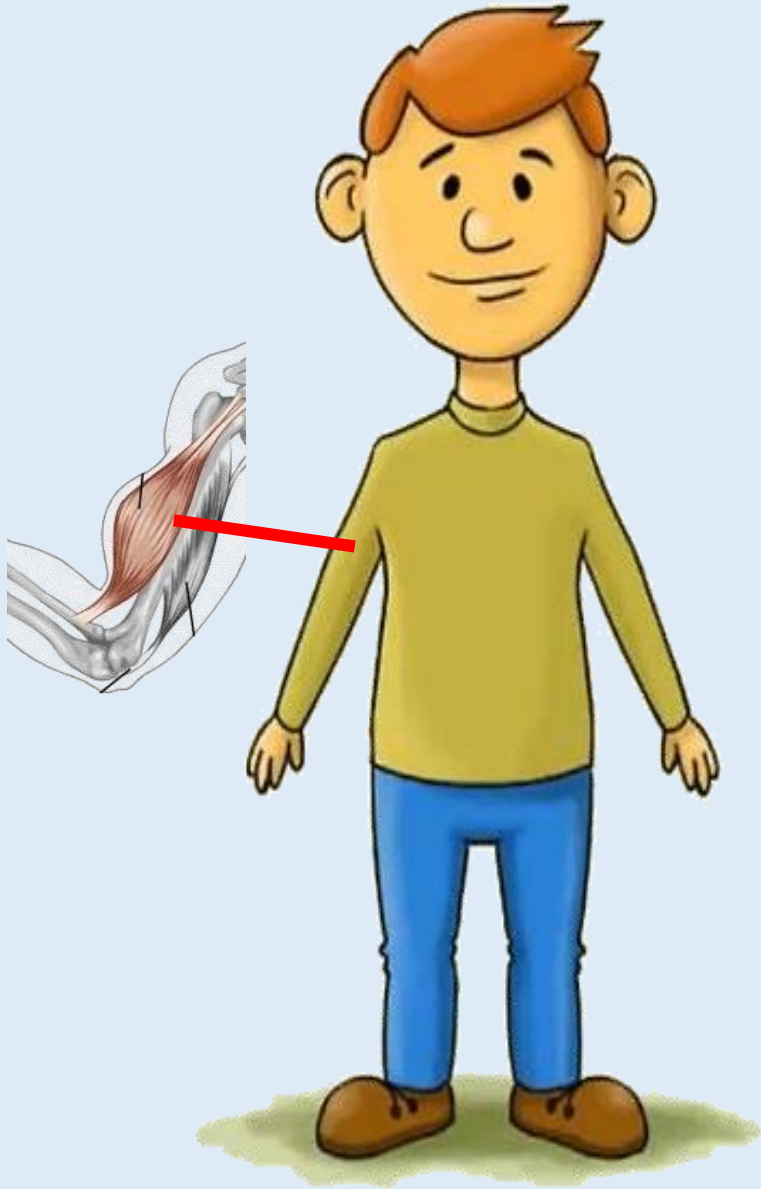
ROBOT  
РОБОТ

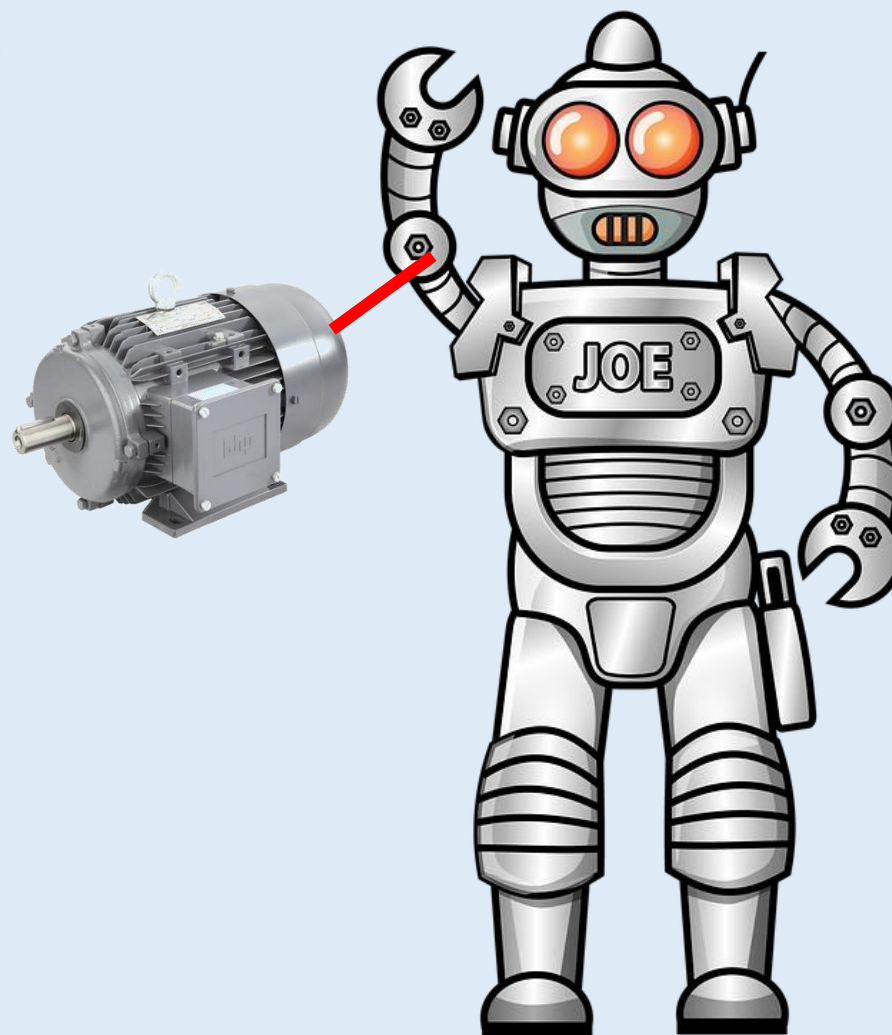


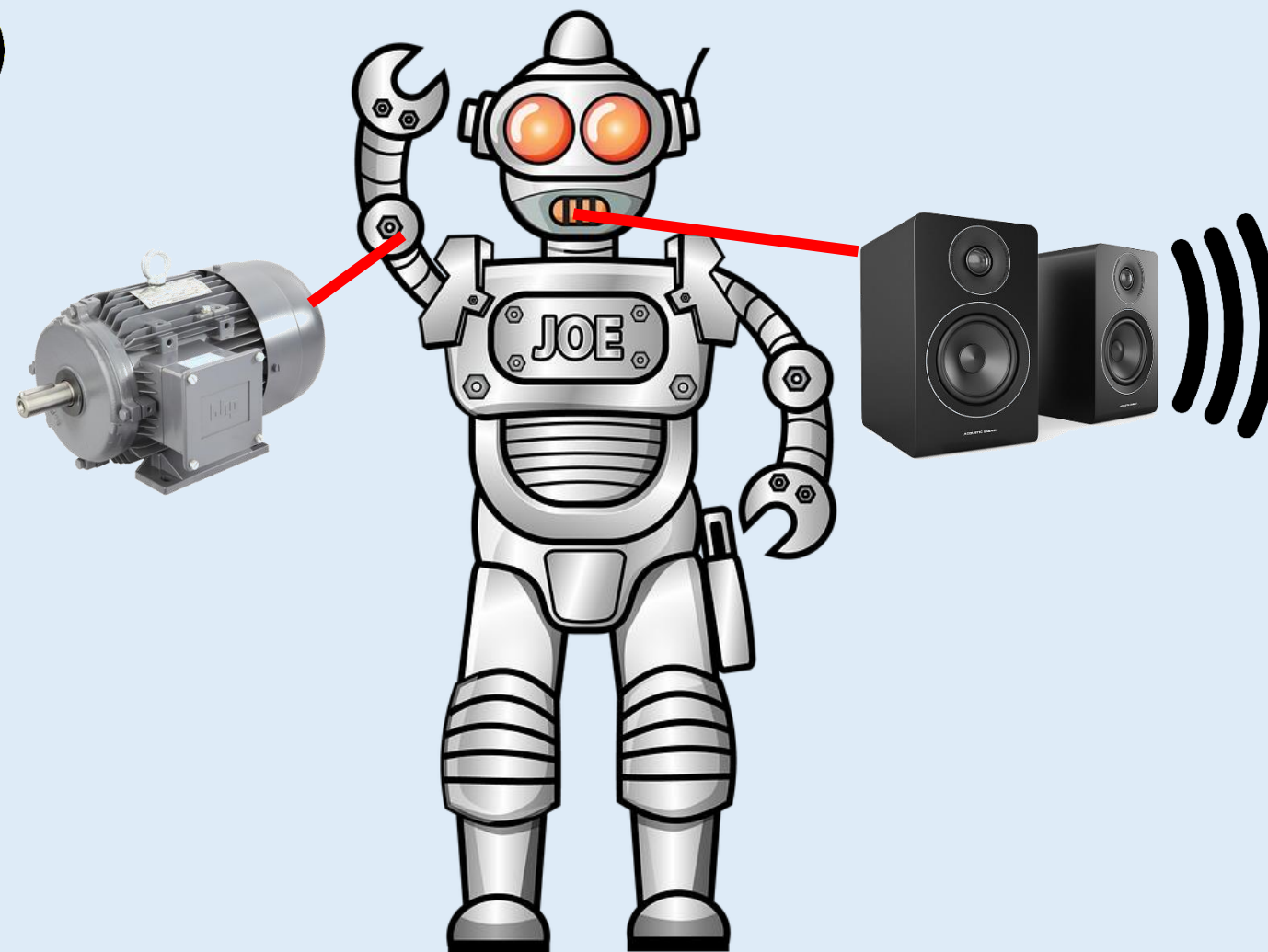


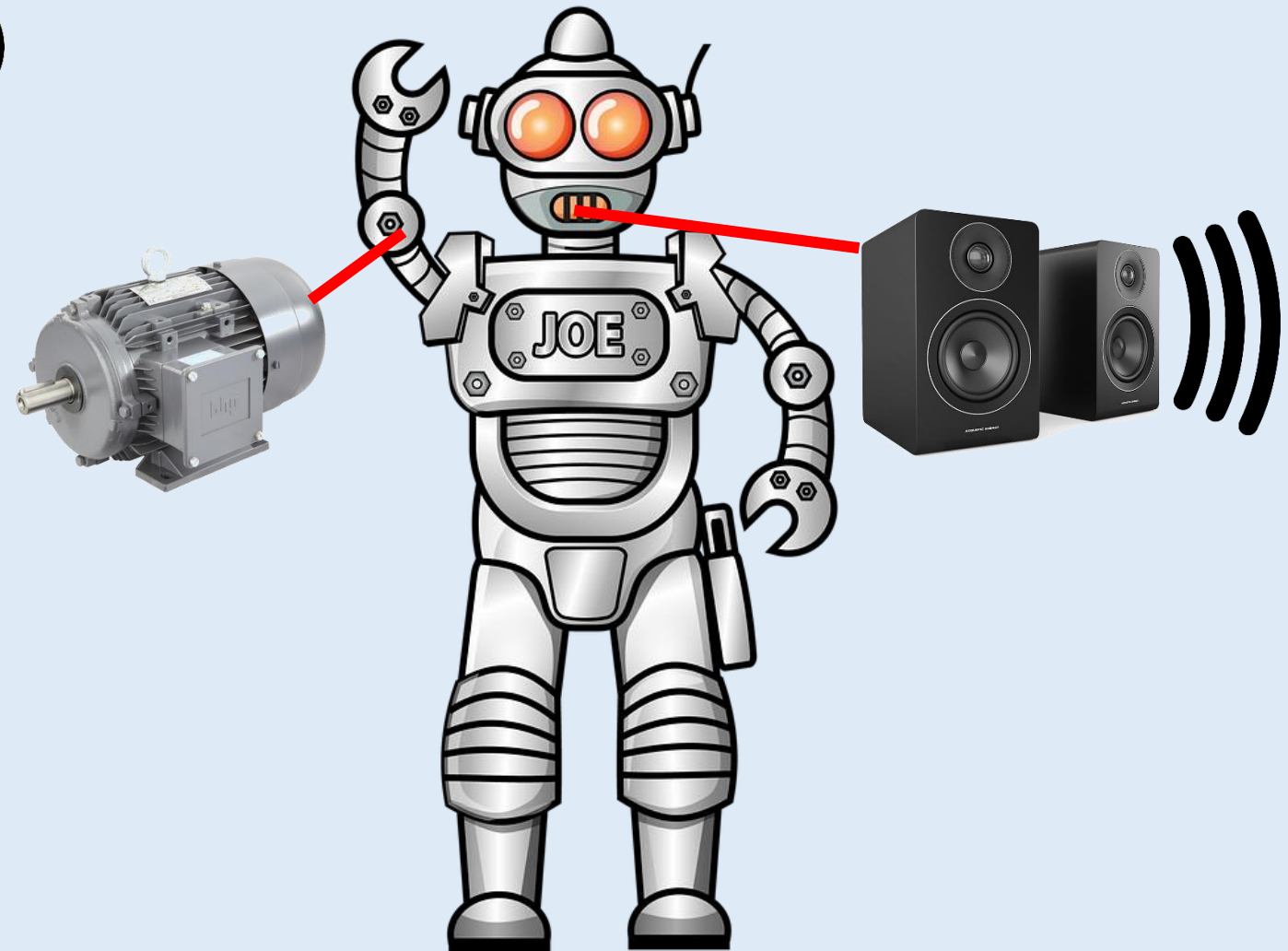
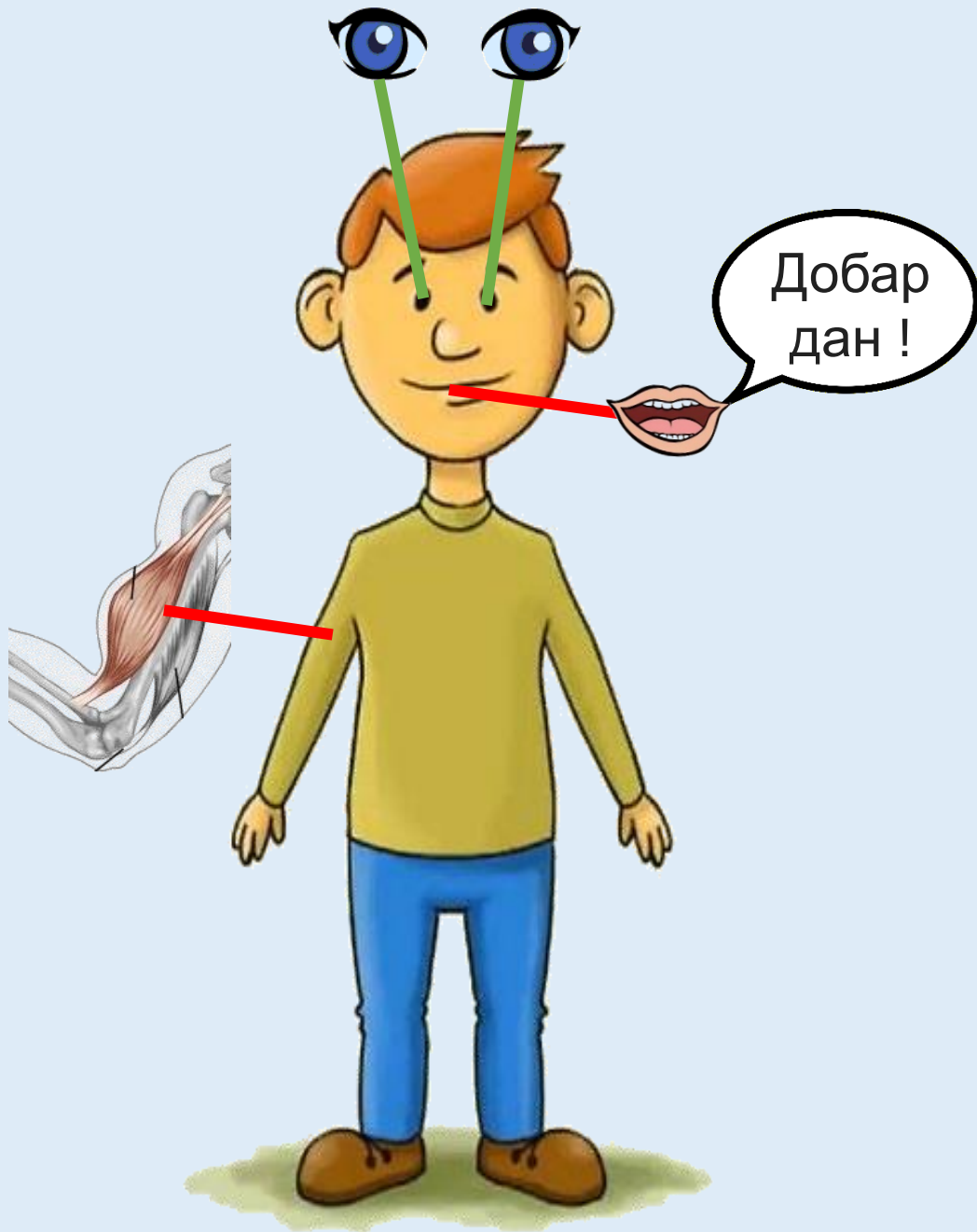


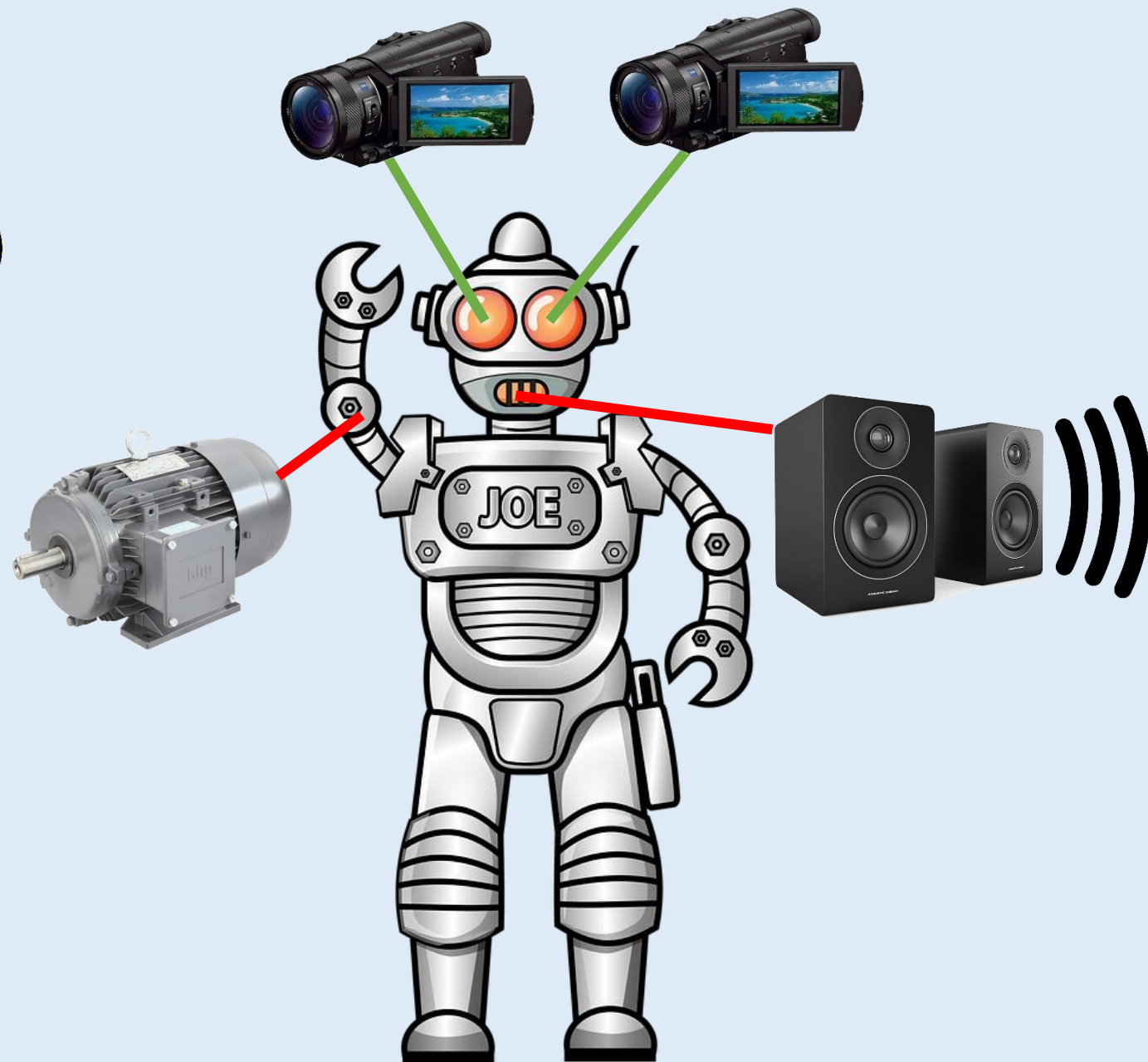
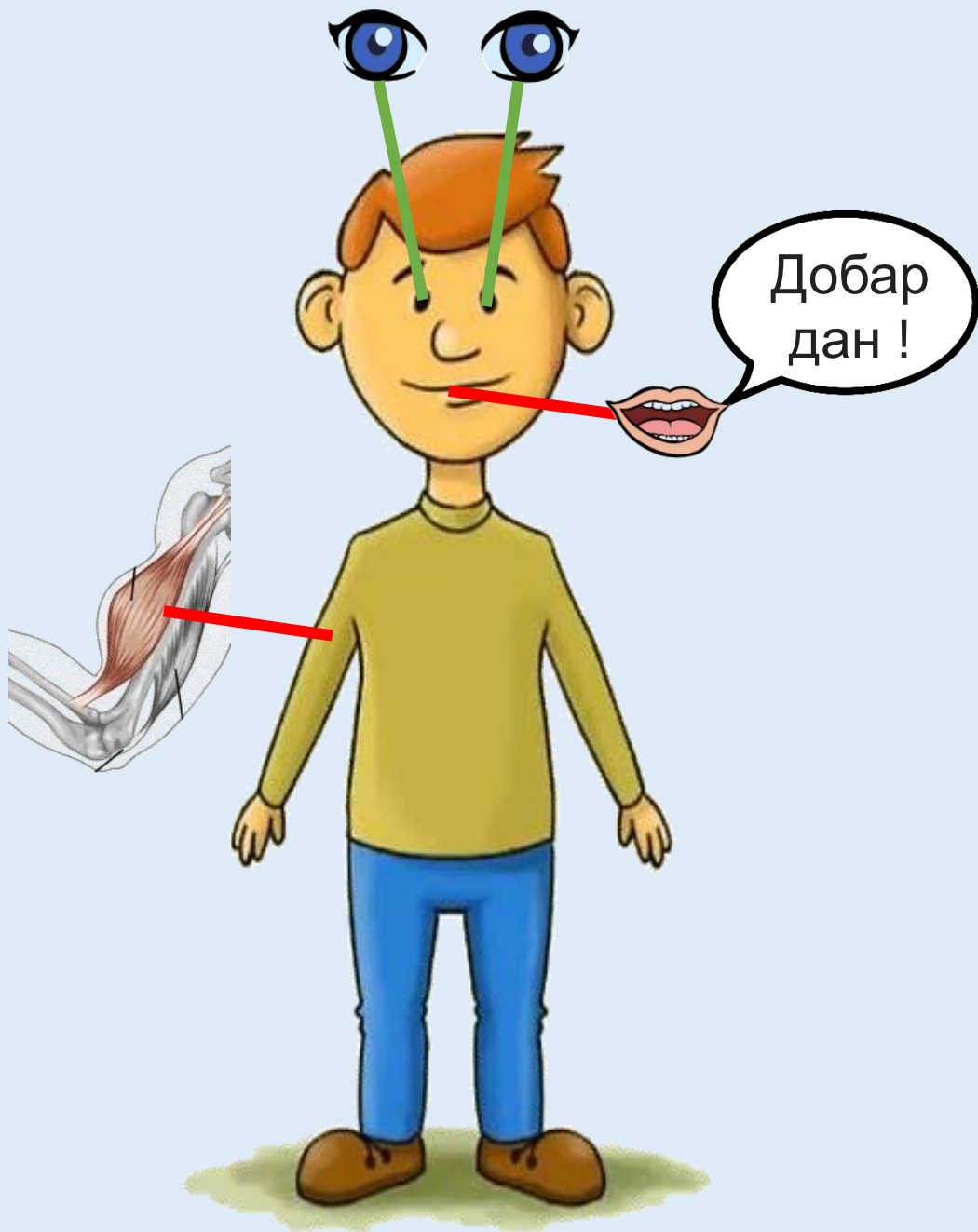


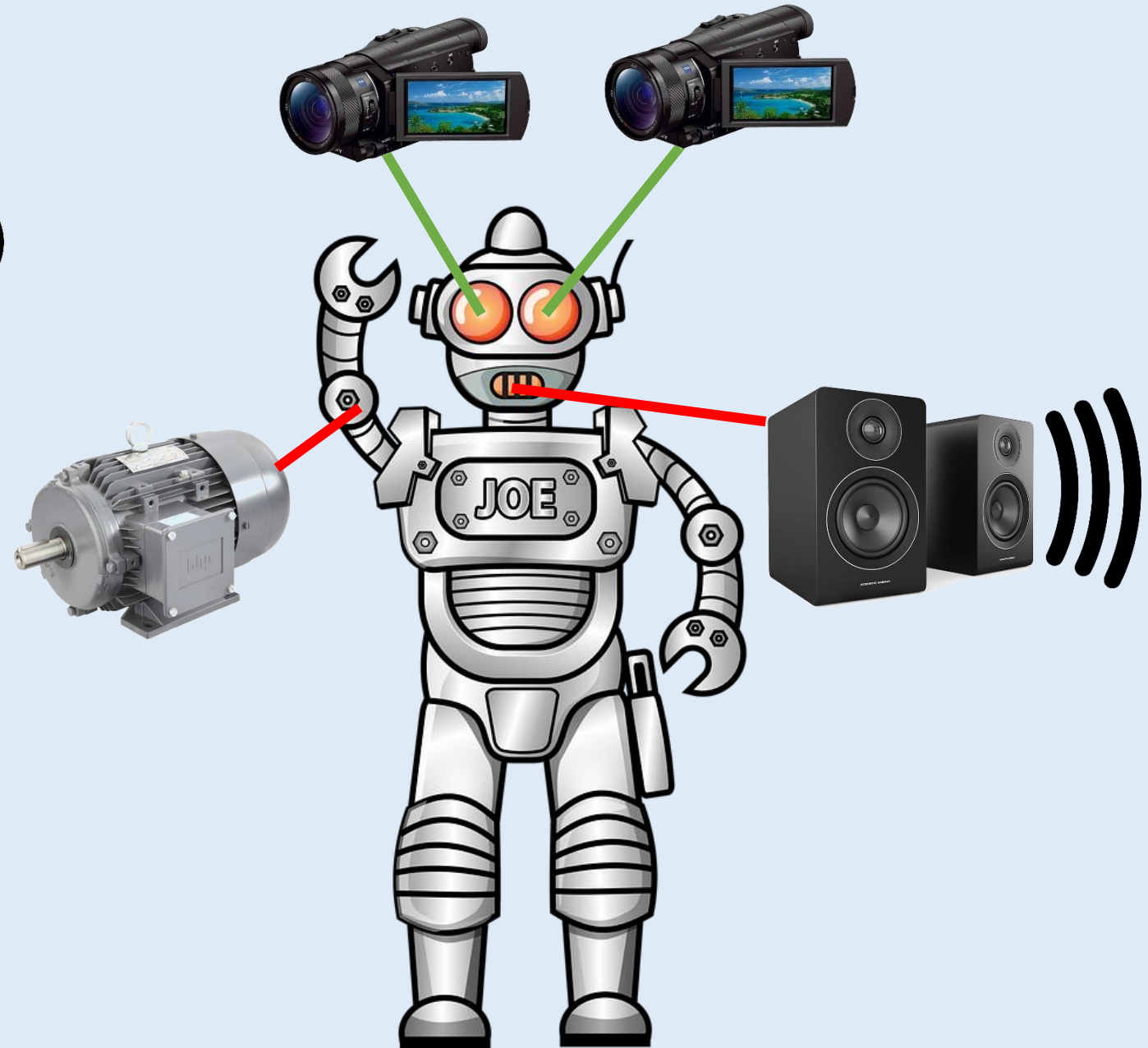
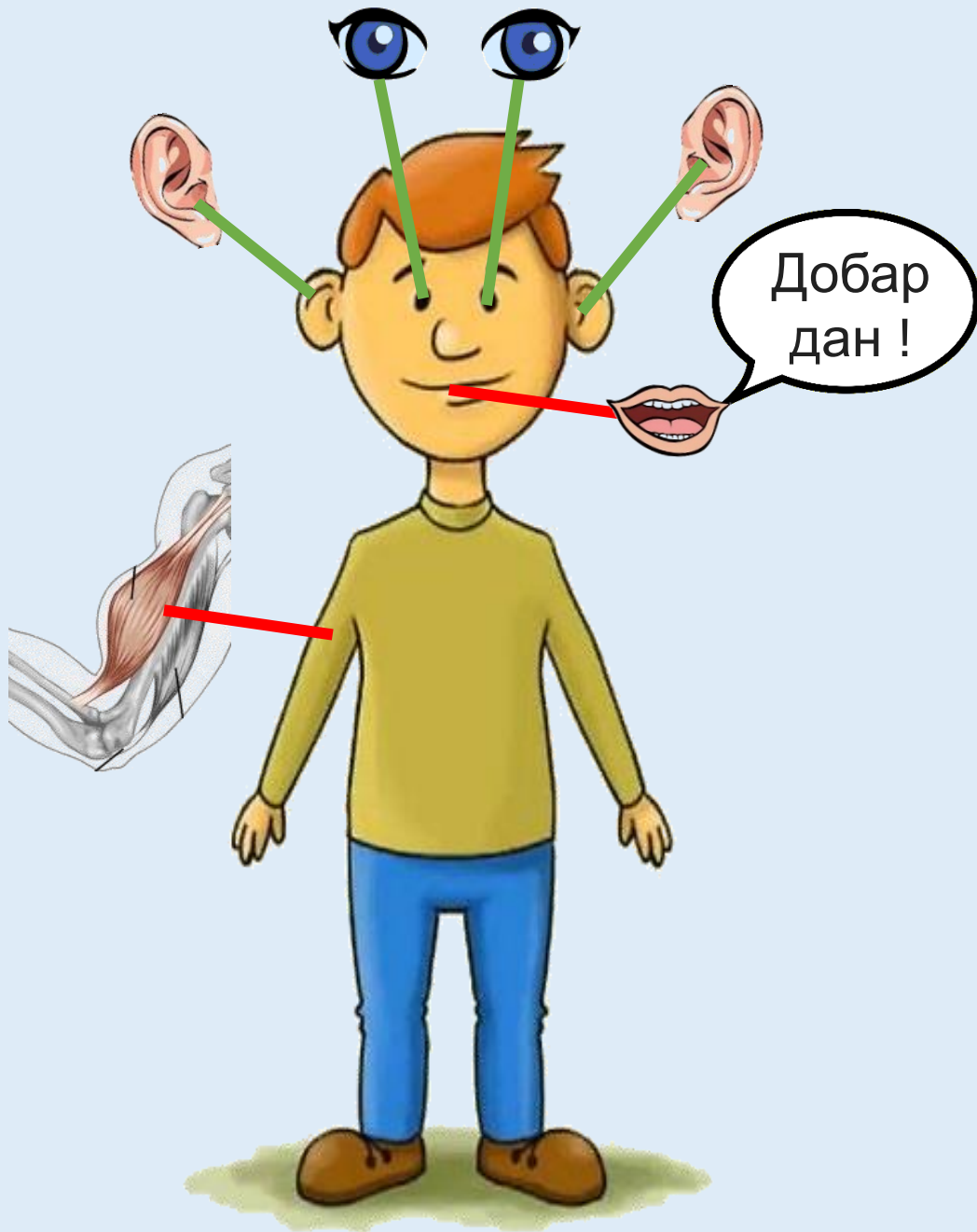


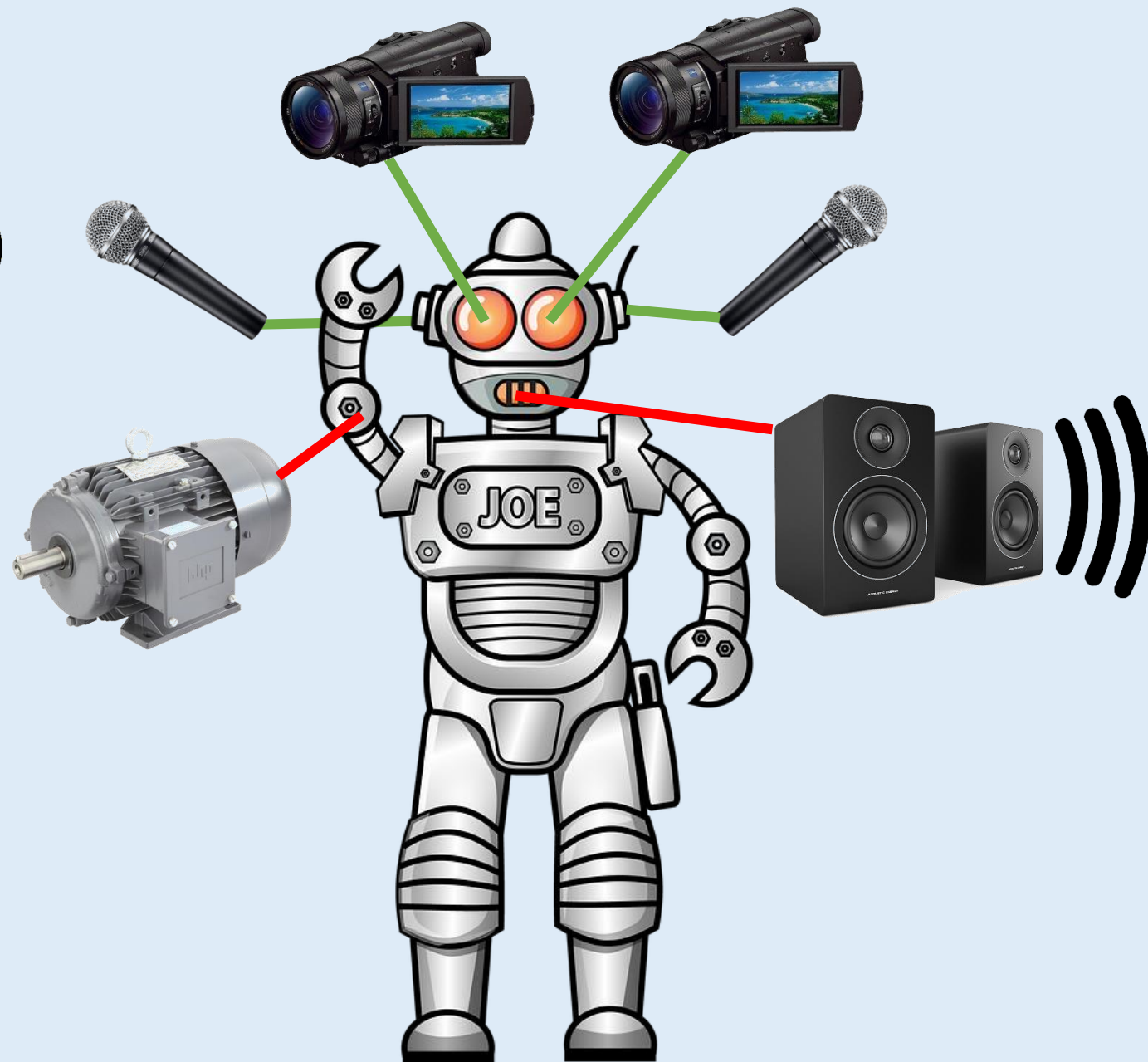
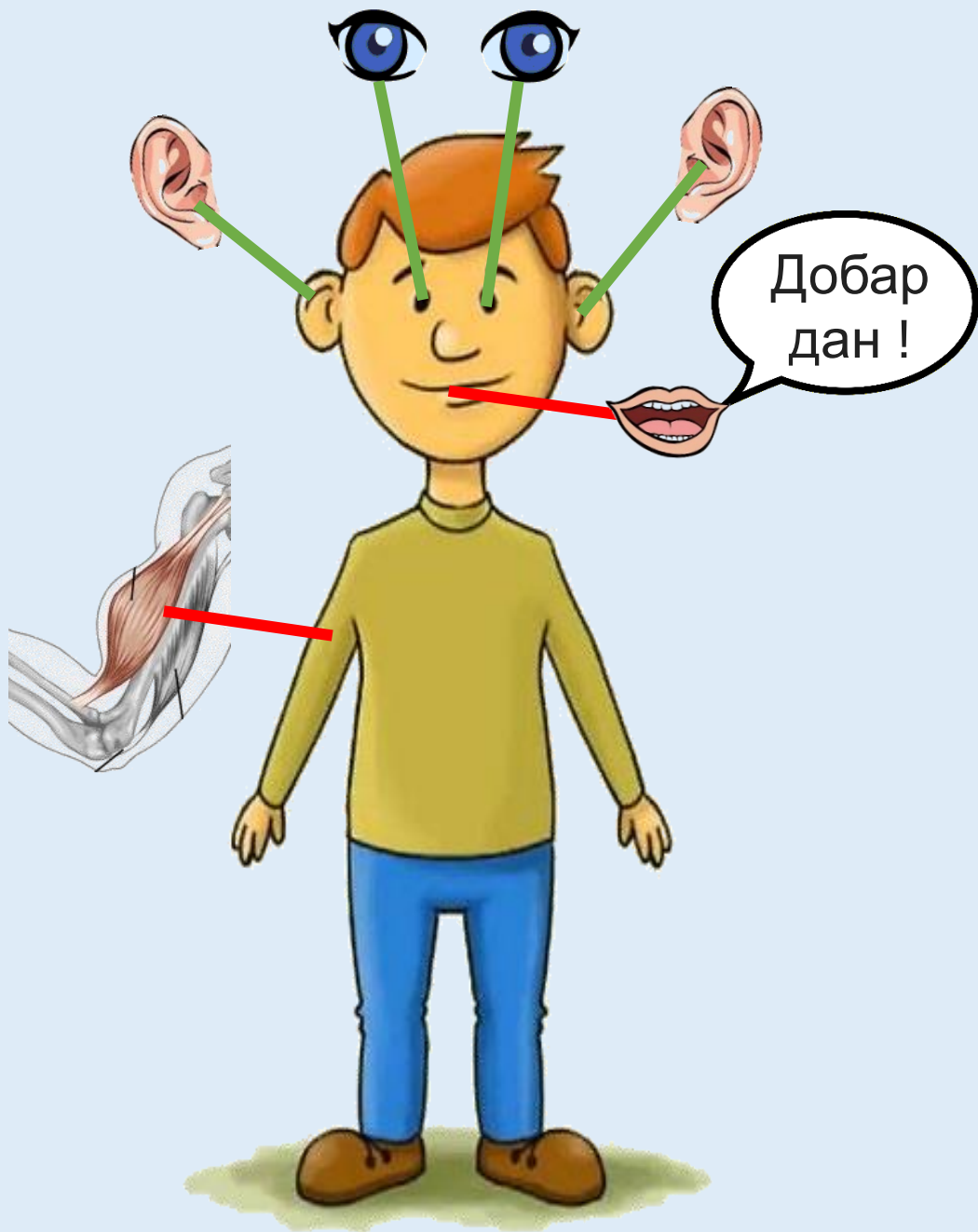


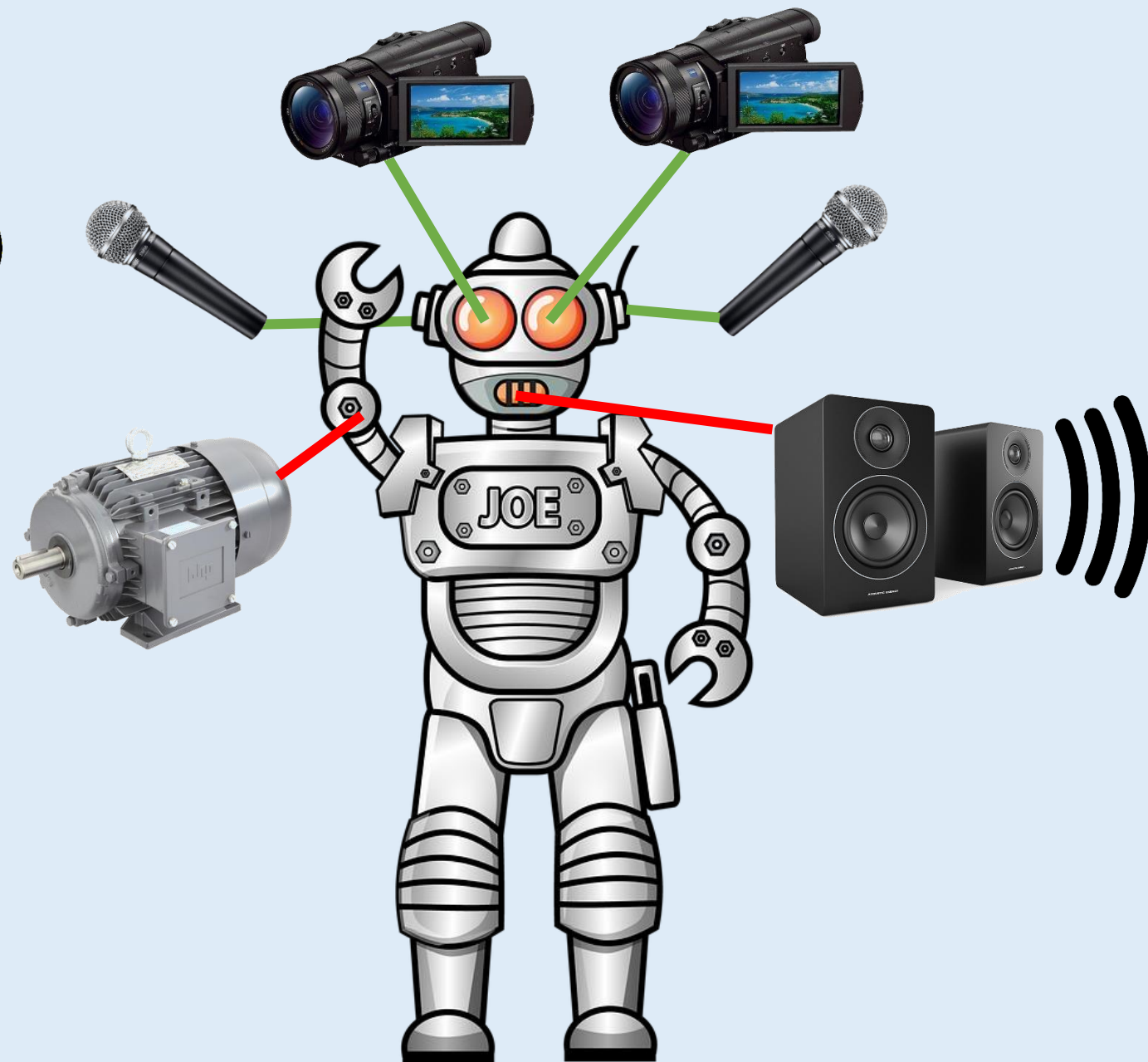


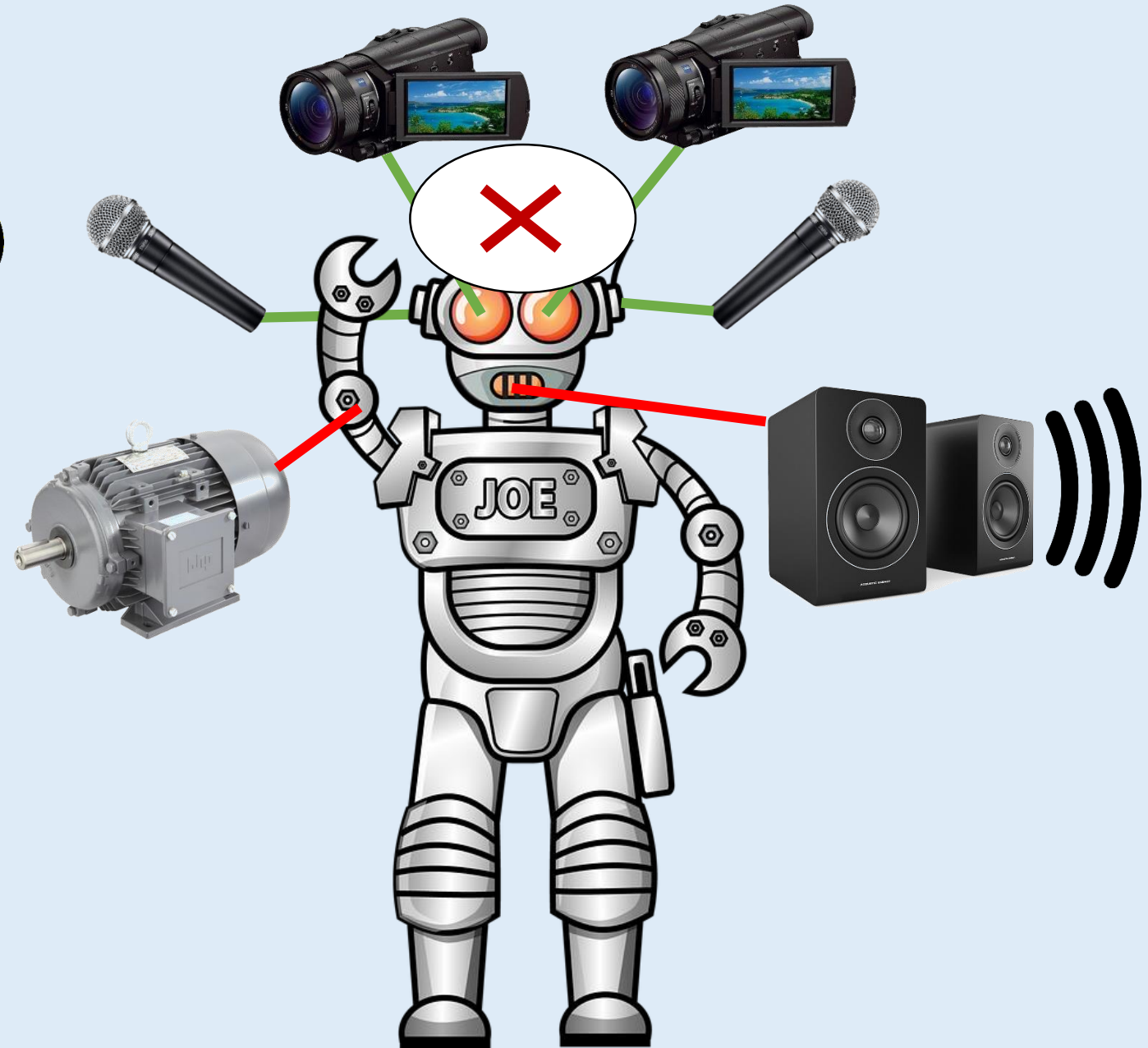


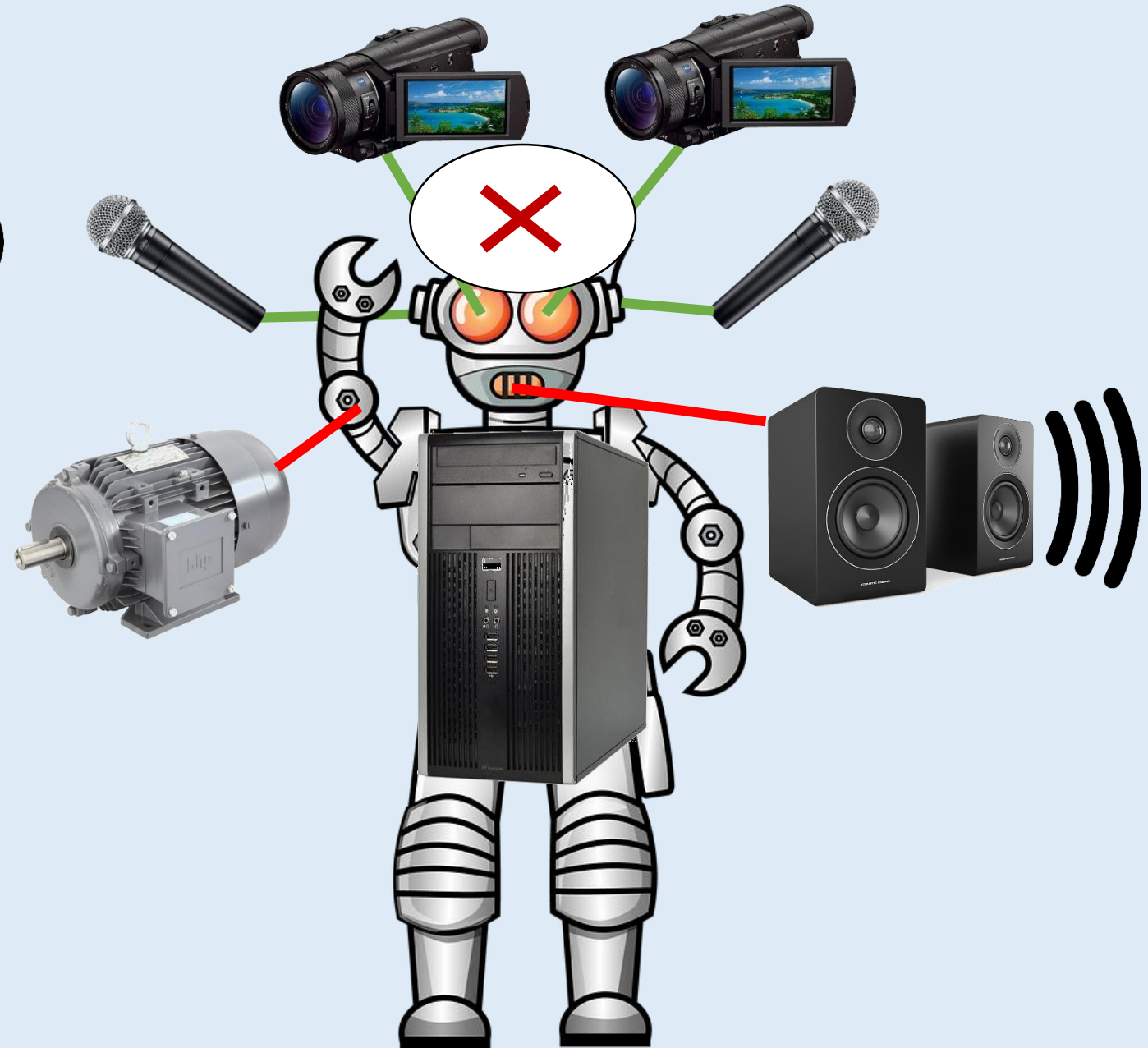
















-[12.57, 3.14, 78.03]  
-3\*8/(457+8/3)^4  
-HELLO HOW? Where?  
-001010111010101011  
010101011111101000  
101001011001000010



-[12.57, 3.14, 78.03]  
-3\*8/(457+8/3)^4  
-HELLO HOW? Where?  
-001010111010101011  
010101011111101000  
101001011001000010

-11111011000110101  
010101000010000101  
010111101001011001  
-125578,83357,42,1  
-[0.00017,4.8795]  
-HELLO, YES

```
base64.cc
31 void base64_encode(const uint8_t * data, size_t len, char * dst)
32 {
33     size_t src_idx = 0;
34     size_t dst_idx = 0;
35     for (; (src_idx + 2) < len; src_idx += 3, dst_idx += 4)
36     {
37         uint8_t s0 = data[src_idx];
38         uint8_t s1 = data[src_idx + 1];
39         uint8_t s2 = data[src_idx + 2];
40
41         dst[dst_idx + 0] = charset[(s0 & 0xfc) >> 2];
42         dst[dst_idx + 1] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
43         dst[dst_idx + 2] = charset[((s1 & 0x0f) << 2) | (s2 & 0xc0) >> 6];
44         dst[dst_idx + 3] = charset[(s2 & 0x3f)];
45     }
46
47     if (src_idx < len)
48     {
49         uint8_t s0 = data[src_idx];
50         uint8_t s1 = (src_idx + 1 < len) ? data[src_idx + 1] : 0;
51
52         dst[dst_idx++] = charset[(s0 & 0xfc) >> 2];
53         dst[dst_idx++] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
54         if (src_idx + 1 < len)
55             dst[dst_idx++] = charset[((s1 & 0x0f) << 2)];
56     }
57 }
```

Line 31, Column 52 Spaces: 4 C++

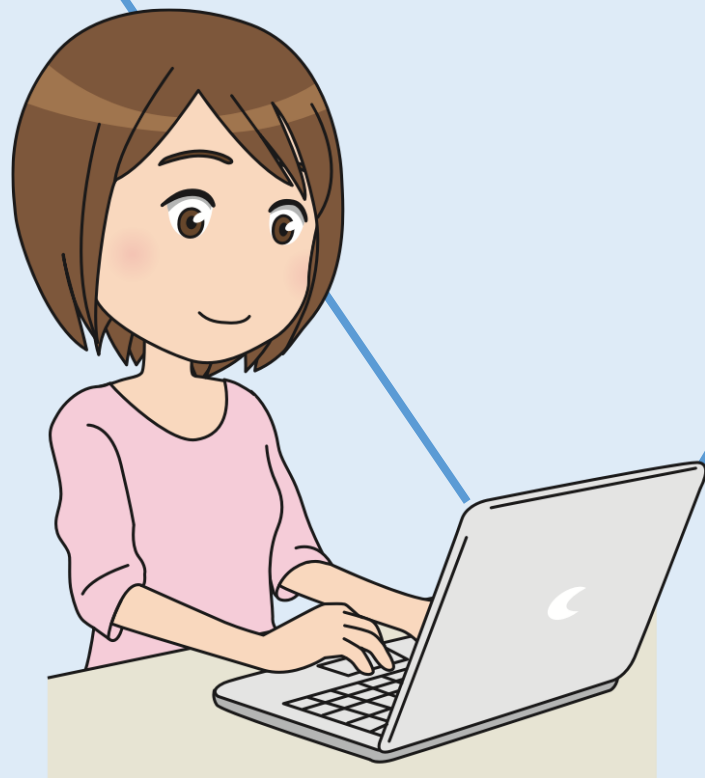


-[12.57, 3.14, 78.03]  
-3\*8/(457+8/3)^4  
-HELLO HOW? Where?  
010101011010101011  
101010111111101000  
101001011001000010

-11111011000110101  
010101000010000101  
010111101001011001  
-125578,83357,42,1  
-[0.00017,4.8795]  
-HELLO, YES

```
base64.cc
31 void base64_encode(const uint8_t * data, size_t len, char * dst)
32 {
33     size_t src_idx = 0;
34     size_t dst_idx = 0;
35     for (; (src_idx + 2) < len; src_idx += 3, dst_idx += 4)
36     {
37         uint8_t s0 = data[src_idx];
38         uint8_t s1 = data[src_idx + 1];
39         uint8_t s2 = data[src_idx + 2];
40
41         dst[dst_idx + 0] = charset[(s0 & 0xfc) >> 2];
42         dst[dst_idx + 1] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
43         dst[dst_idx + 2] = charset[((s1 & 0x0f) << 2) | (s2 & 0xc0) >> 6];
44         dst[dst_idx + 3] = charset[(s2 & 0x3f)];
45     }
46
47     if (src_idx < len)
48     {
49         uint8_t s0 = data[src_idx];
50         uint8_t s1 = (src_idx + 1 < len) ? data[src_idx + 1] : 0;
51
52         dst[dst_idx++] = charset[(s0 & 0xfc) >> 2];
53         dst[dst_idx++] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
54         if (src_idx + 1 < len)
55             dst[dst_idx++] = charset[((s1 & 0x0f) << 2)];
56     }
57 }
```

Line 31, Column 52      Spaces: 4      C++

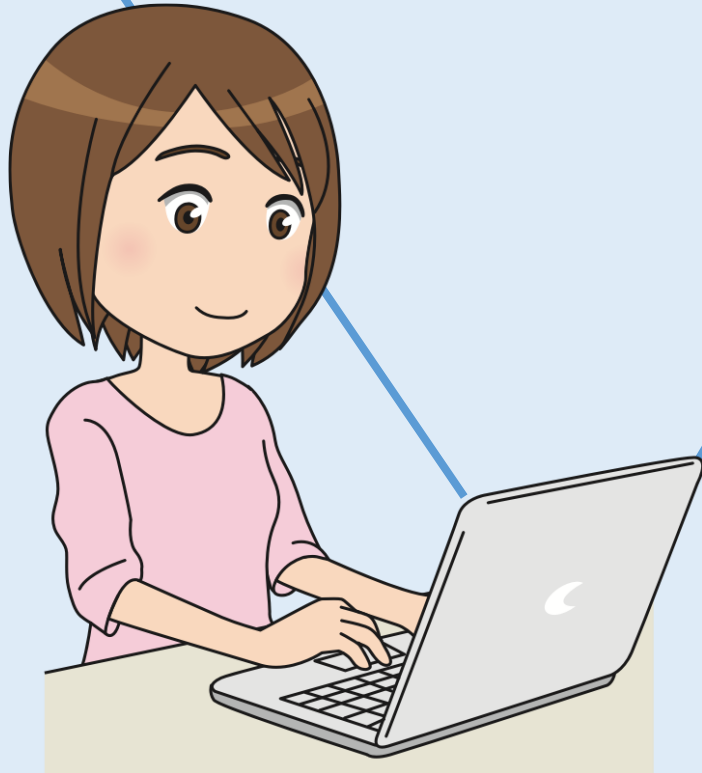


-[12.57, 3.14, 78.03]  
-3\*8/(457+8/3)^4  
-HELLO HOW? Where?  
010101011010101011  
101010111111101000  
101001011001000010

-11111011000110101  
01010100000100000101  
010111101001011001  
-125578,83357,42,1  
-[0.00017,4.8795]  
-HELLO, YES

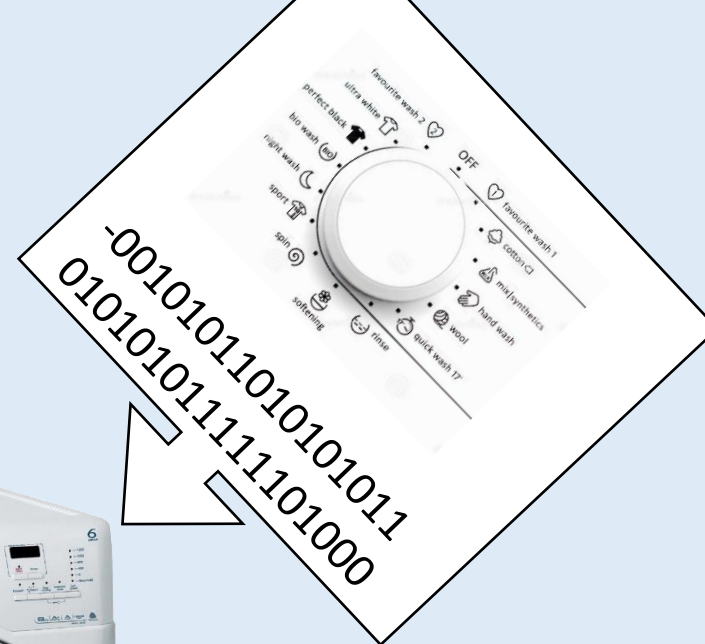
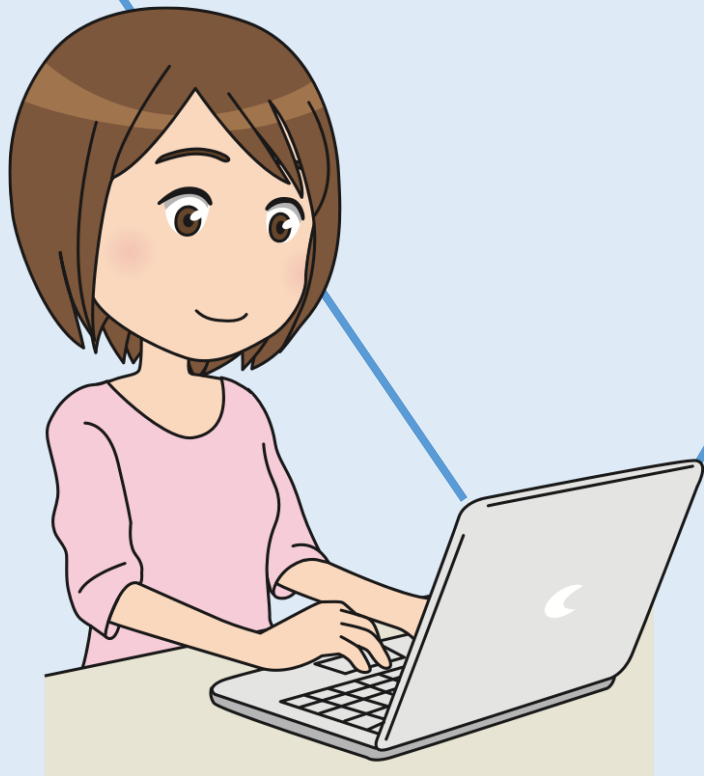
```
base64.cc
31 void base64_encode(const uint8_t * data, size_t len, char * dst)
32 {
33     size_t src_idx = 0;
34     size_t dst_idx = 0;
35     for (; (src_idx + 2) < len; src_idx += 3, dst_idx += 4)
36     {
37         uint8_t s0 = data[src_idx];
38         uint8_t s1 = data[src_idx + 1];
39         uint8_t s2 = data[src_idx + 2];
40
41         dst[dst_idx + 0] = charset[(s0 & 0xfc) >> 2];
42         dst[dst_idx + 1] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
43         dst[dst_idx + 2] = charset[((s1 & 0x0f) << 2) | (s2 & 0xc0) >> 6];
44         dst[dst_idx + 3] = charset[(s2 & 0x3f)];
45     }
46
47     if (src_idx < len)
48     {
49         uint8_t s0 = data[src_idx];
50         uint8_t s1 = (src_idx + 1 < len) ? data[src_idx + 1] : 0;
51
52         dst[dst_idx++] = charset[(s0 & 0xfc) >> 2];
53         dst[dst_idx++] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
54         if (src_idx + 1 < len)
55             dst[dst_idx++] = charset[((s1 & 0x0f) << 2)];
56     }
57 }
```

FileHippo.com  
Line 31, Column 52  
Spaces: 4  
C++



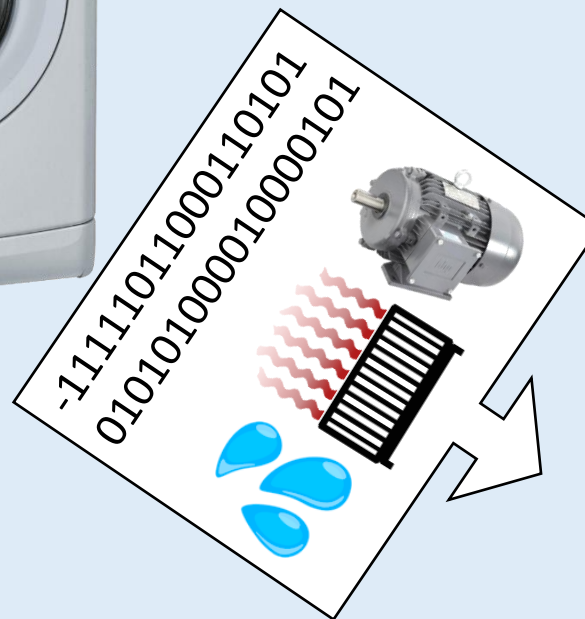
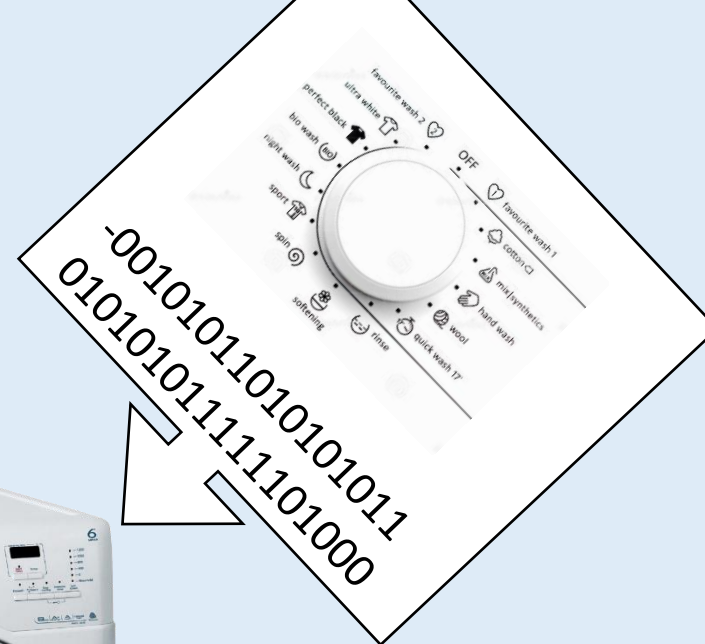
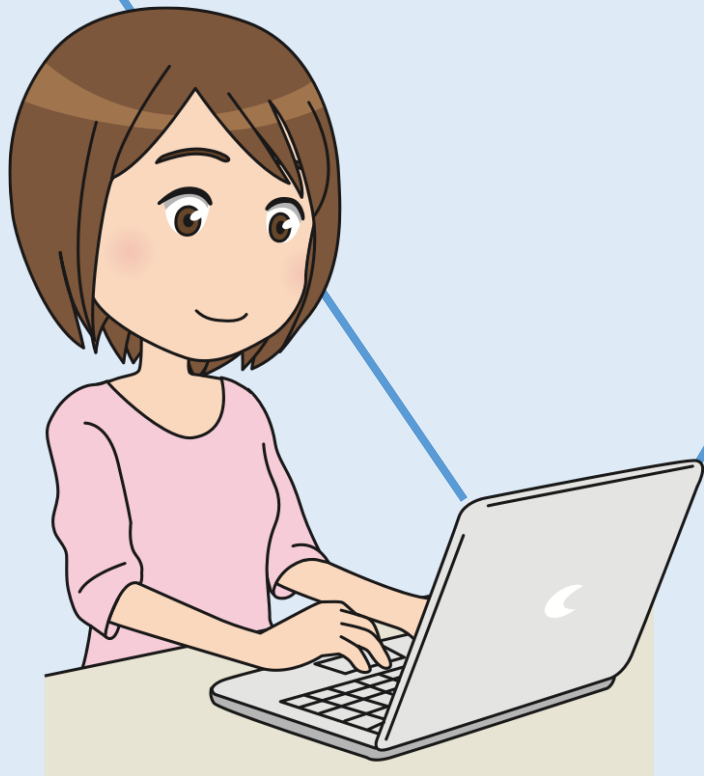
```
base64.cc
31 void base64_encode(const uint8_t * data, size_t len, char * dst)
32 {
33     size_t src_idx = 0;
34     size_t dst_idx = 0;
35     for (; (src_idx + 2) < len; src_idx += 3, dst_idx += 4)
36     {
37         uint8_t s0 = data[src_idx];
38         uint8_t s1 = data[src_idx + 1];
39         uint8_t s2 = data[src_idx + 2];
40
41         dst[dst_idx + 0] = charset[(s0 & 0xfc) >> 2];
42         dst[dst_idx + 1] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
43         dst[dst_idx + 2] = charset[((s1 & 0x0f) << 2) | (s2 & 0xc0) >> 6];
44         dst[dst_idx + 3] = charset[(s2 & 0x3f)];
45     }
46
47     if (src_idx < len)
48     {
49         uint8_t s0 = data[src_idx];
50         uint8_t s1 = (src_idx + 1 < len) ? data[src_idx + 1] : 0;
51
52         dst[dst_idx++] = charset[(s0 & 0xfc) >> 2];
53         dst[dst_idx++] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
54         if (src_idx + 1 < len)
55             dst[dst_idx++] = charset[((s1 & 0x0f) << 2)];
56     }
57 }
```

FileHippo.com  
Line 31, Column 52  
Spaces: 4  
C++

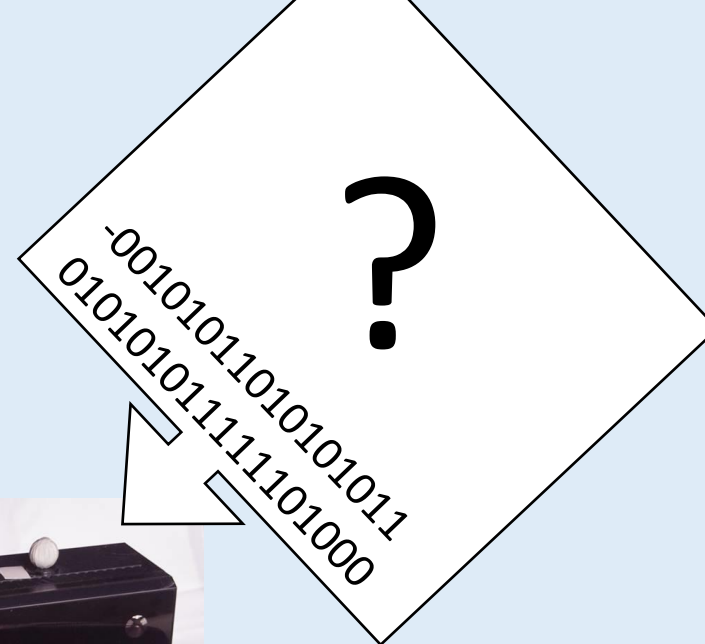
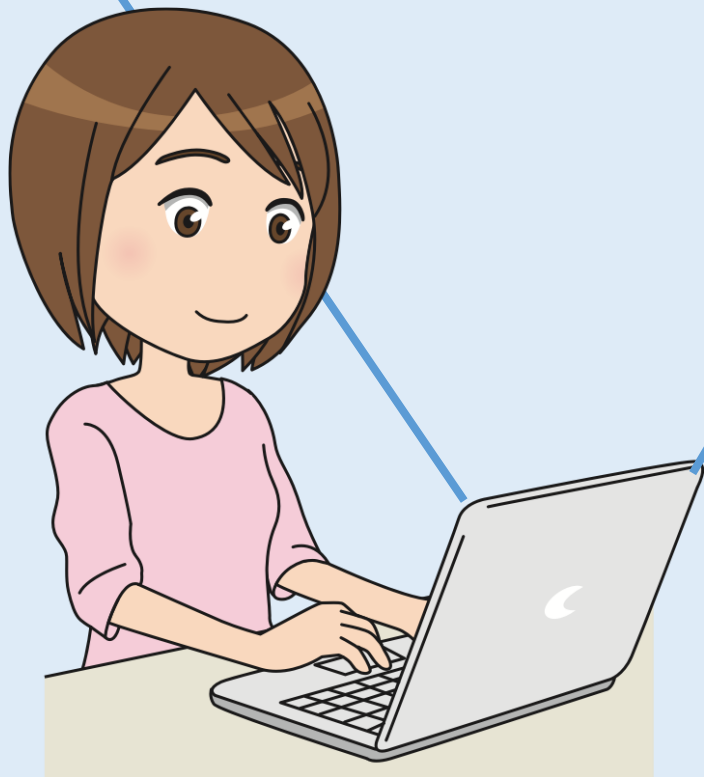


```
base64.cc
31 void base64_encode(const uint8_t * data, size_t len, char * dst)
32 {
33     size_t src_idx = 0;
34     size_t dst_idx = 0;
35     for (; (src_idx + 2) < len; src_idx += 3, dst_idx += 4)
36     {
37         uint8_t s0 = data[src_idx];
38         uint8_t s1 = data[src_idx + 1];
39         uint8_t s2 = data[src_idx + 2];
40
41         dst[dst_idx + 0] = charset[(s0 & 0xfc) >> 2];
42         dst[dst_idx + 1] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
43         dst[dst_idx + 2] = charset[((s1 & 0x0f) << 2) | (s2 & 0xc0) >> 6];
44         dst[dst_idx + 3] = charset[(s2 & 0x3f)];
45     }
46
47     if (src_idx < len)
48     {
49         uint8_t s0 = data[src_idx];
50         uint8_t s1 = (src_idx + 1 < len) ? data[src_idx + 1] : 0;
51
52         dst[dst_idx++] = charset[(s0 & 0xfc) >> 2];
53         dst[dst_idx++] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
54         if (src_idx + 1 < len)
55             dst[dst_idx++] = charset[((s1 & 0x0f) << 2)];
56     }
57 }
```

Line 31, Column 52      Spaces: 4      C++



```
base64.cc
31 void base64_encode(const uint8_t * data, size_t len, char * dst)
32 {
33     size_t src_idx = 0;
34     size_t dst_idx = 0;
35     for (; (src_idx + 2) < len; src_idx += 3, dst_idx += 4)
36     {
37         uint8_t s0 = data[src_idx];
38         uint8_t s1 = data[src_idx + 1];
39         uint8_t s2 = data[src_idx + 2];
40
41         dst[dst_idx + 0] = charset[(s0 & 0xfc) >> 2];
42         dst[dst_idx + 1] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
43         dst[dst_idx + 2] = charset[((s1 & 0x0f) << 2) | (s2 & 0xc0) >> 6];
44         dst[dst_idx + 3] = charset[(s2 & 0x3f)];
45     }
46
47     if (src_idx < len)
48     {
49         uint8_t s0 = data[src_idx];
50         uint8_t s1 = (src_idx + 1 < len) ? data[src_idx + 1] : 0;
51
52         dst[dst_idx++] = charset[(s0 & 0xfc) >> 2];
53         dst[dst_idx++] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
54         if (src_idx + 1 < len)
55             dst[dst_idx++] = charset[((s1 & 0x0f) << 2)];
56     }
57 }
```



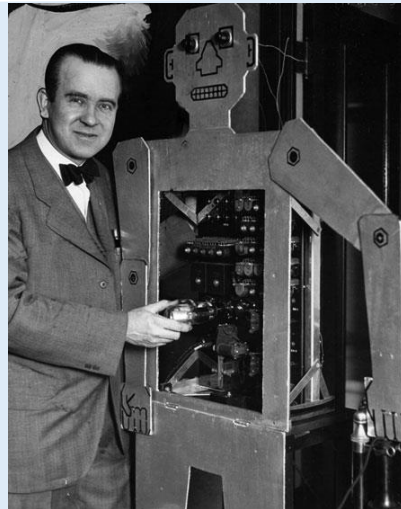
The background of the image is a dark blue gradient. It features several abstract, glowing blue geometric shapes, including circles and polygons, some of which are interconnected by thin white lines. In the upper center, there are two large, stylized gear-like structures. The one on the left is a complex, multi-layered circular design with various internal patterns. The one on the right is a simpler, more circular gear-like structure. To the right of these gears, there is a network of white lines and dots, resembling a circuit board or a data network. A large, white, curved arrow starts from the bottom left and points towards the top right, passing behind the text. The text "ARTIFICIAL INTELLIGENCE" is written in a bold, white, sans-serif font, centered horizontally across the middle of the image.

# ARTIFICIAL INTELLIGENCE

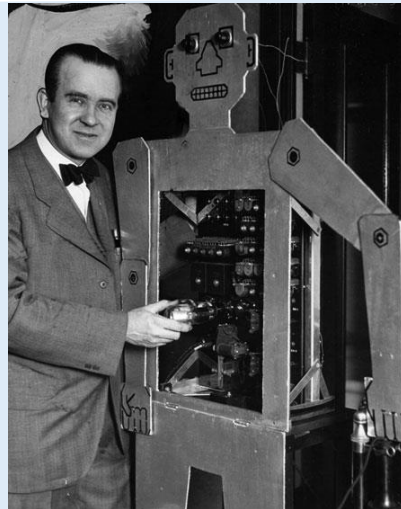
# ARTIFICIAL INTELLIGENCE



# ARTIFICIAL INTELLIGENCE



# ARTIFICIAL INTELLIGENCE



# ARTIFICIAL INTELLIGENCE

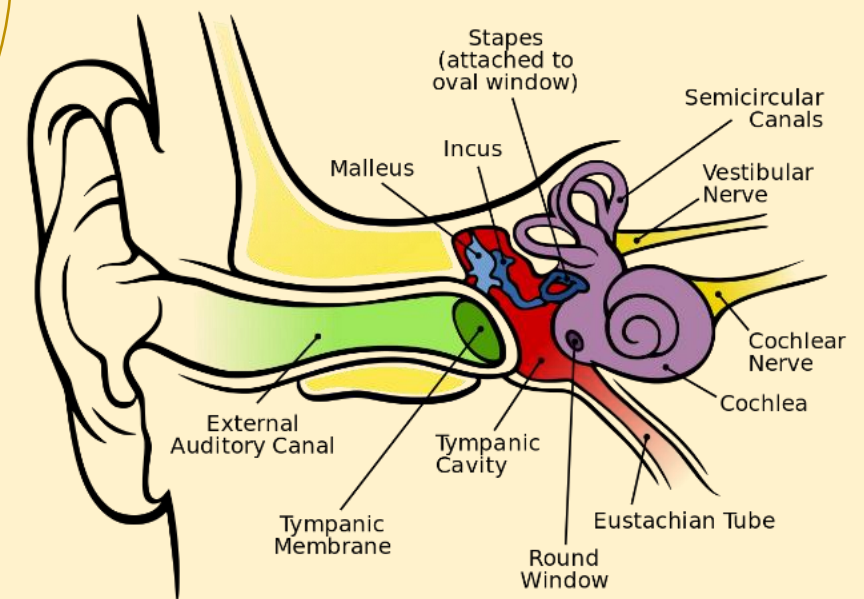
























Speed ~ 350 m/s





Speed  $\sim 350$  m/s



300 000 000 m/s





1, 2, 3...

Distance = Speed x Time

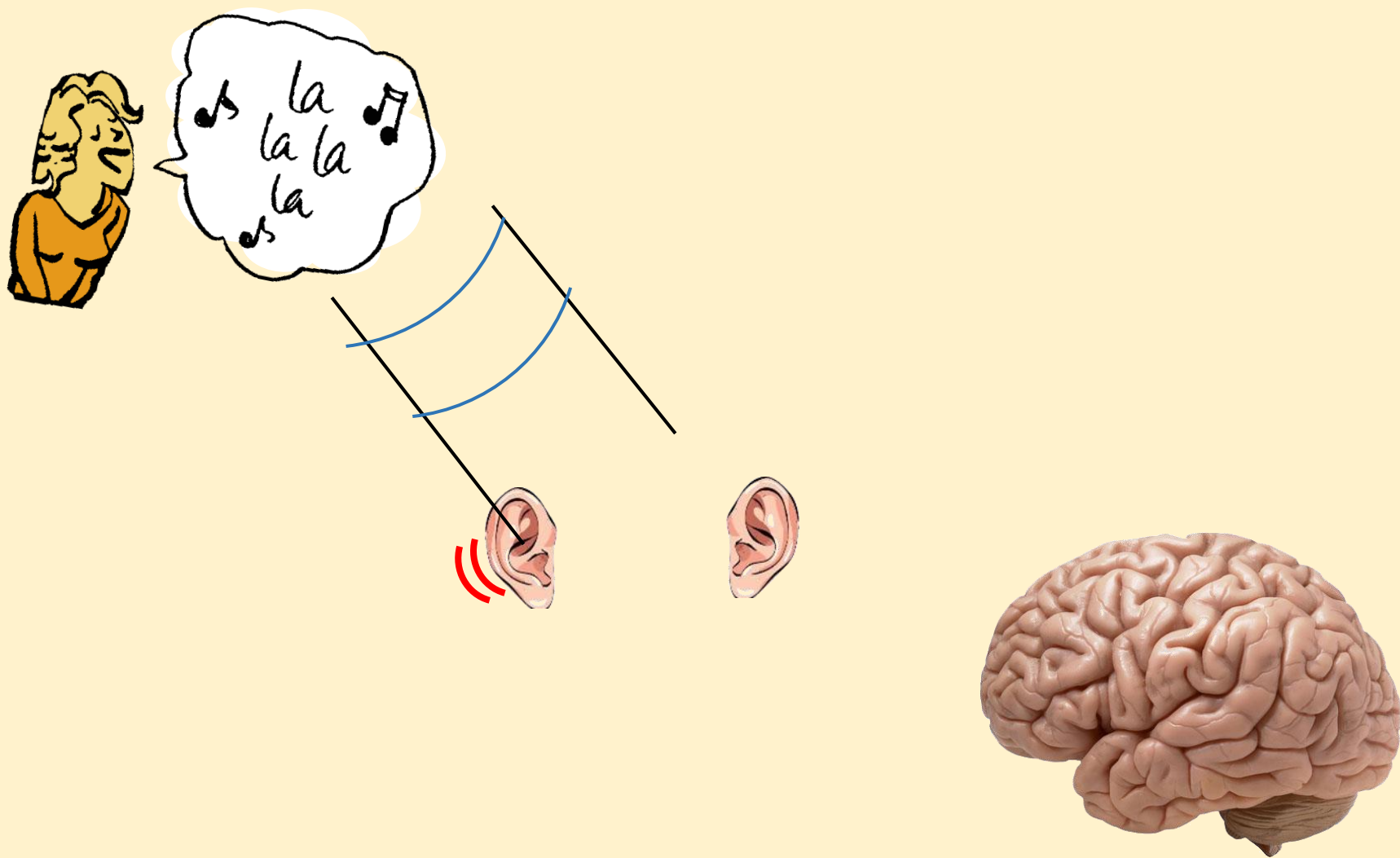
Speed ~ 350 m/s

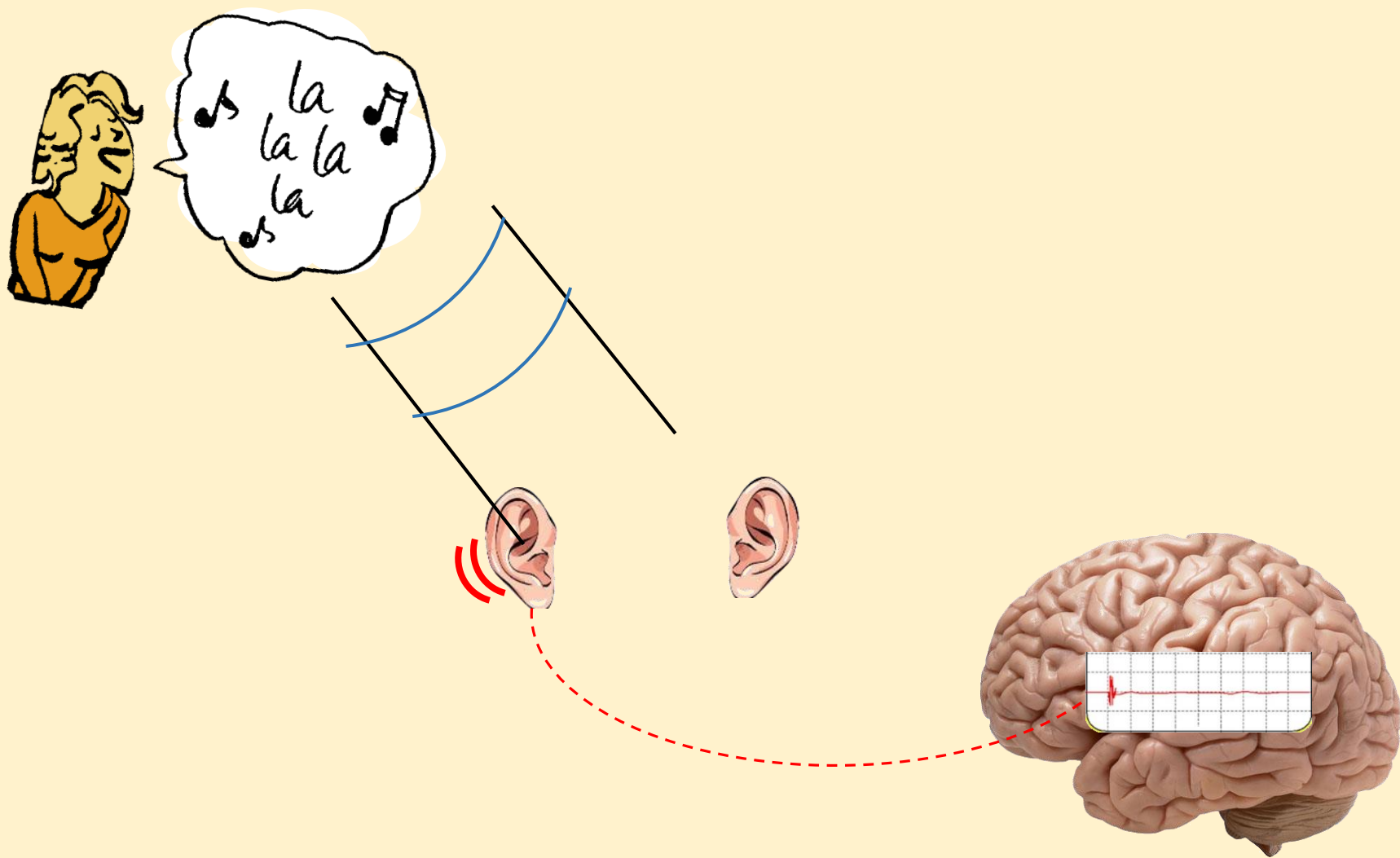


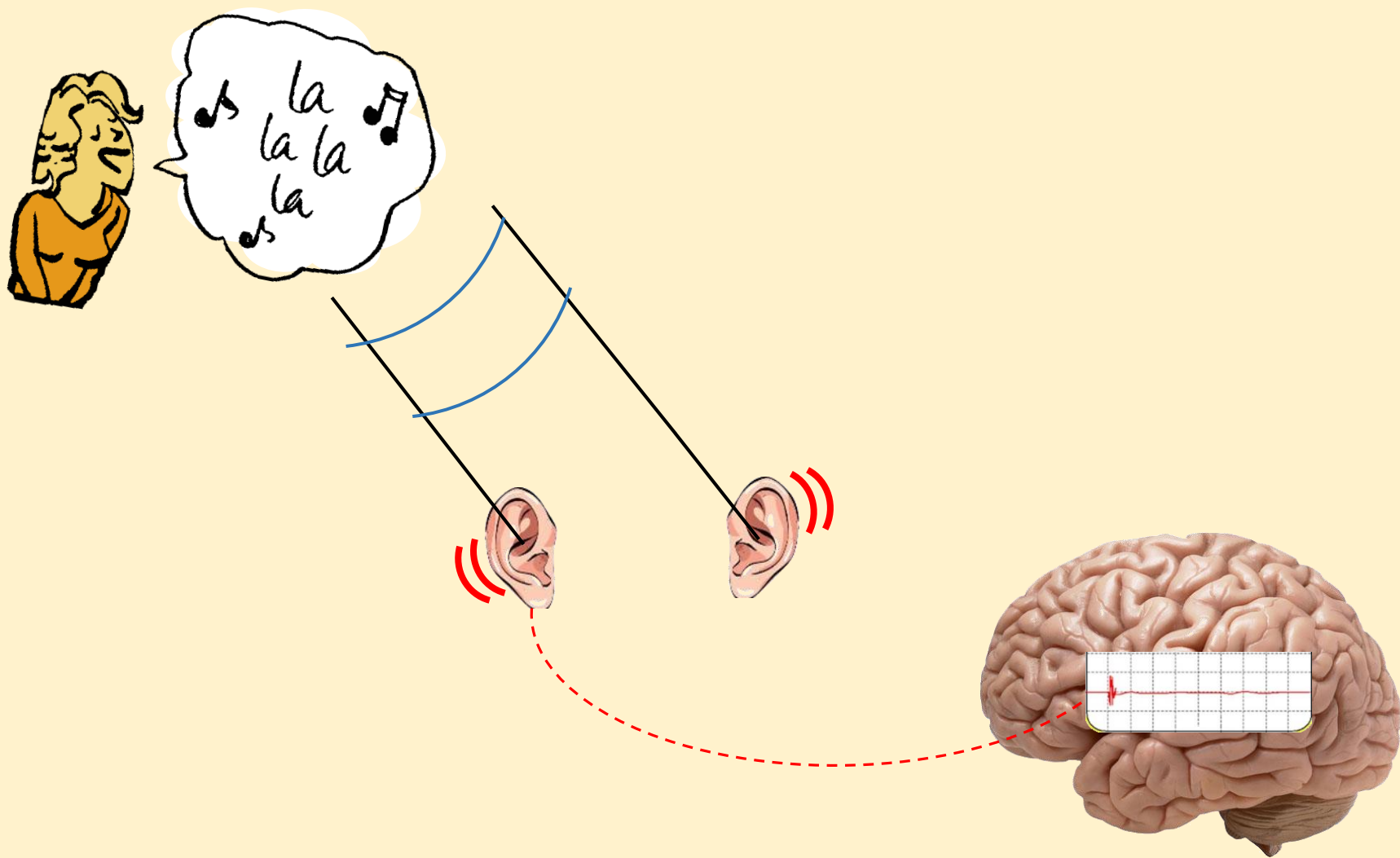
300 000 000 m/s

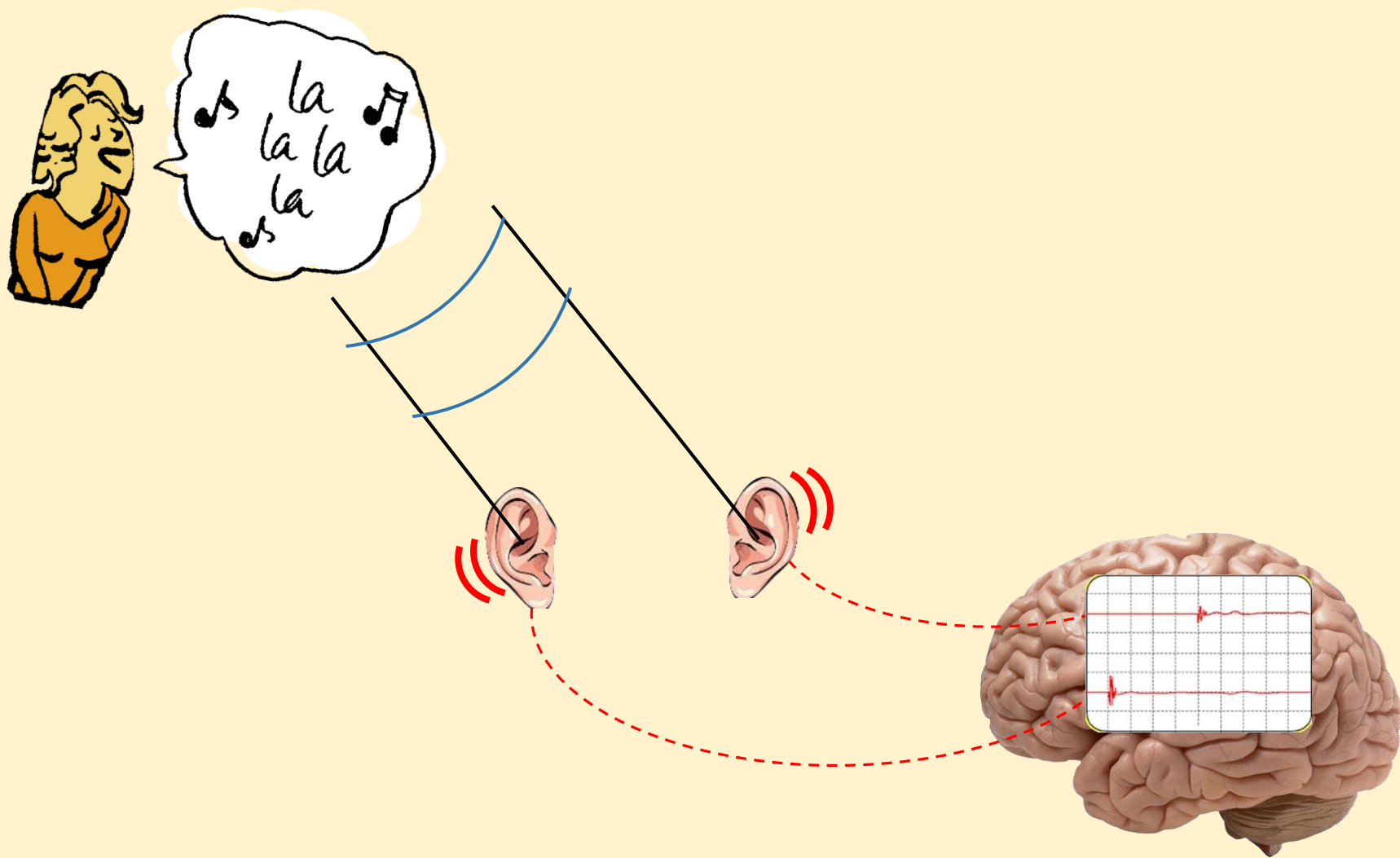


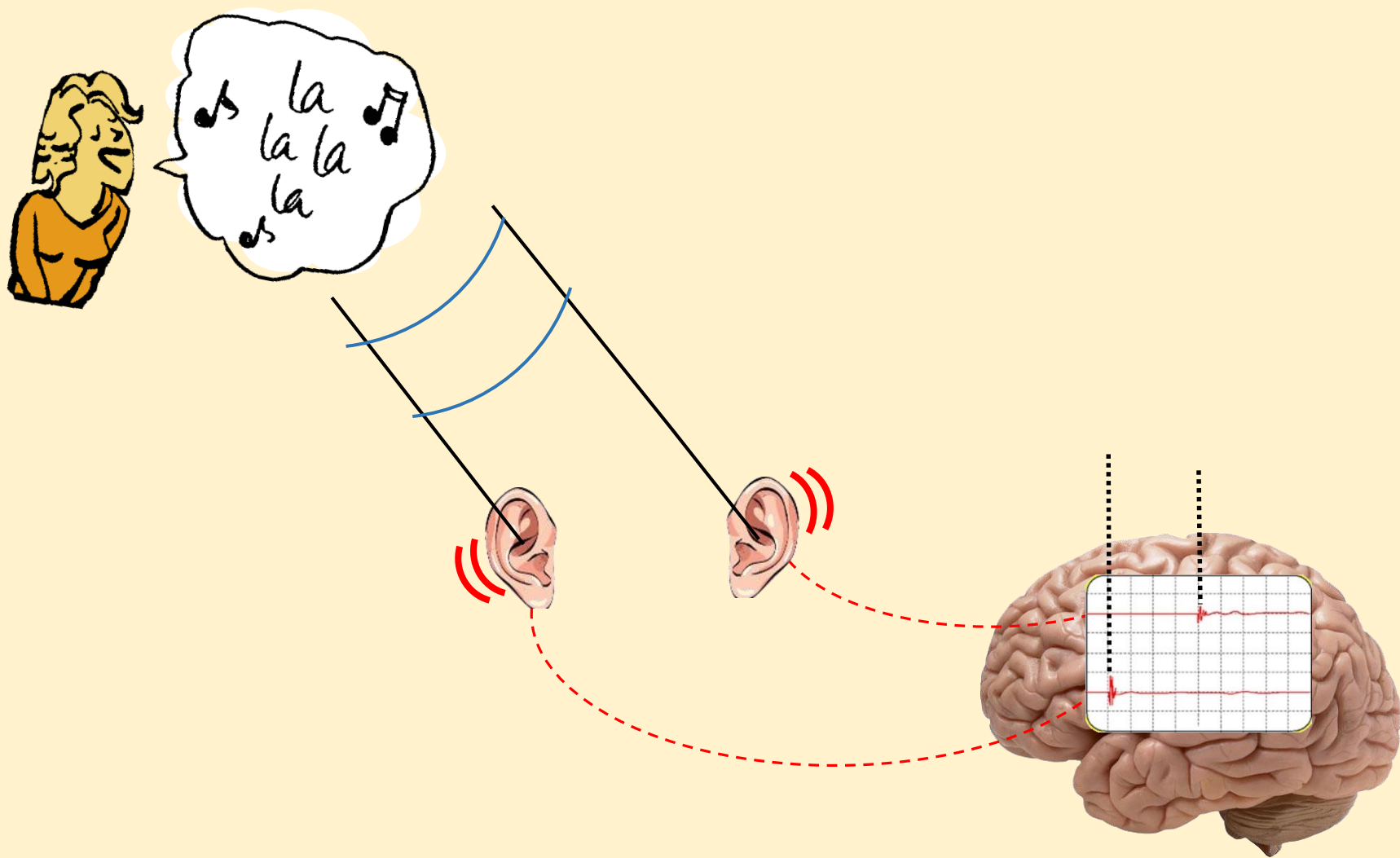


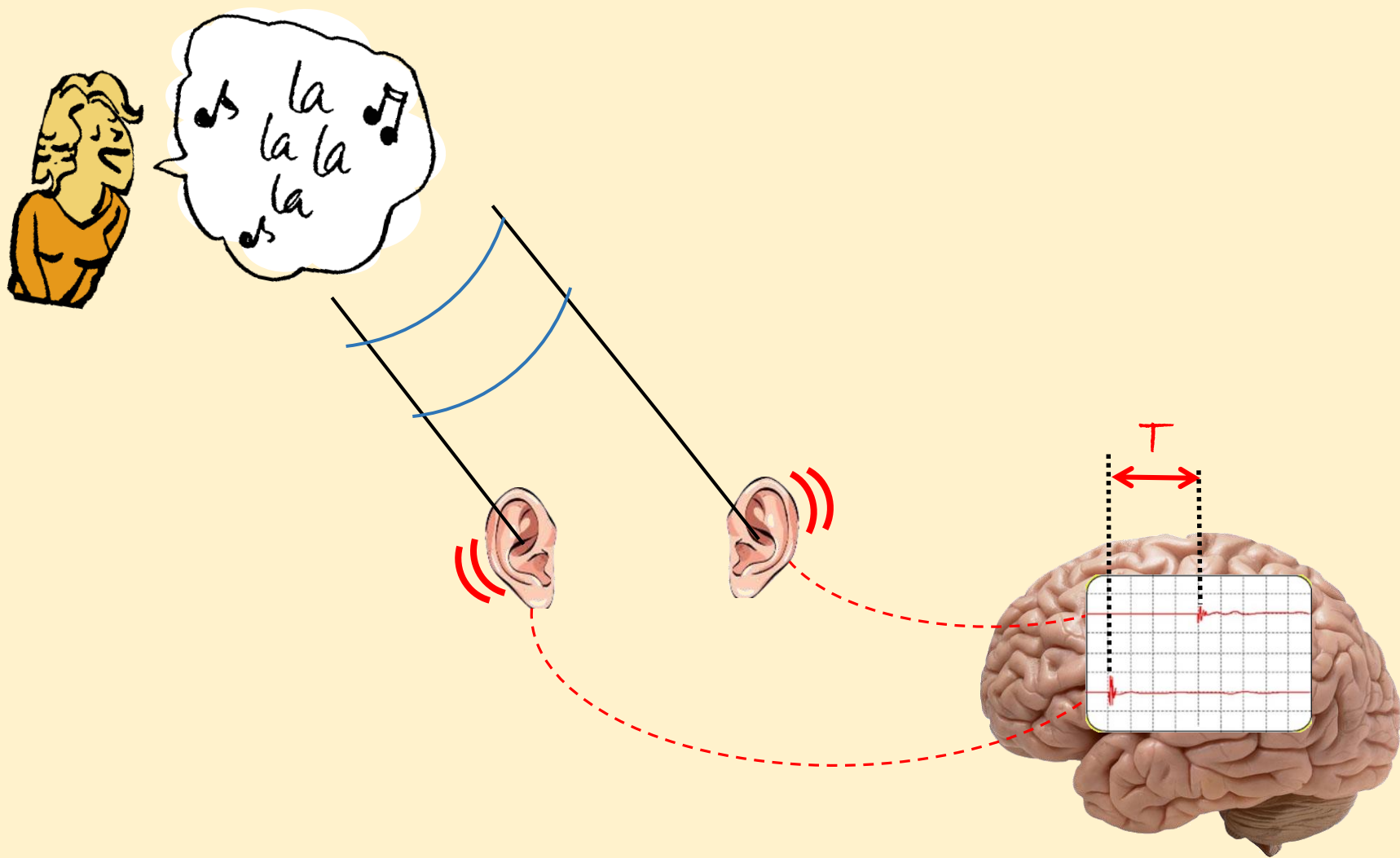


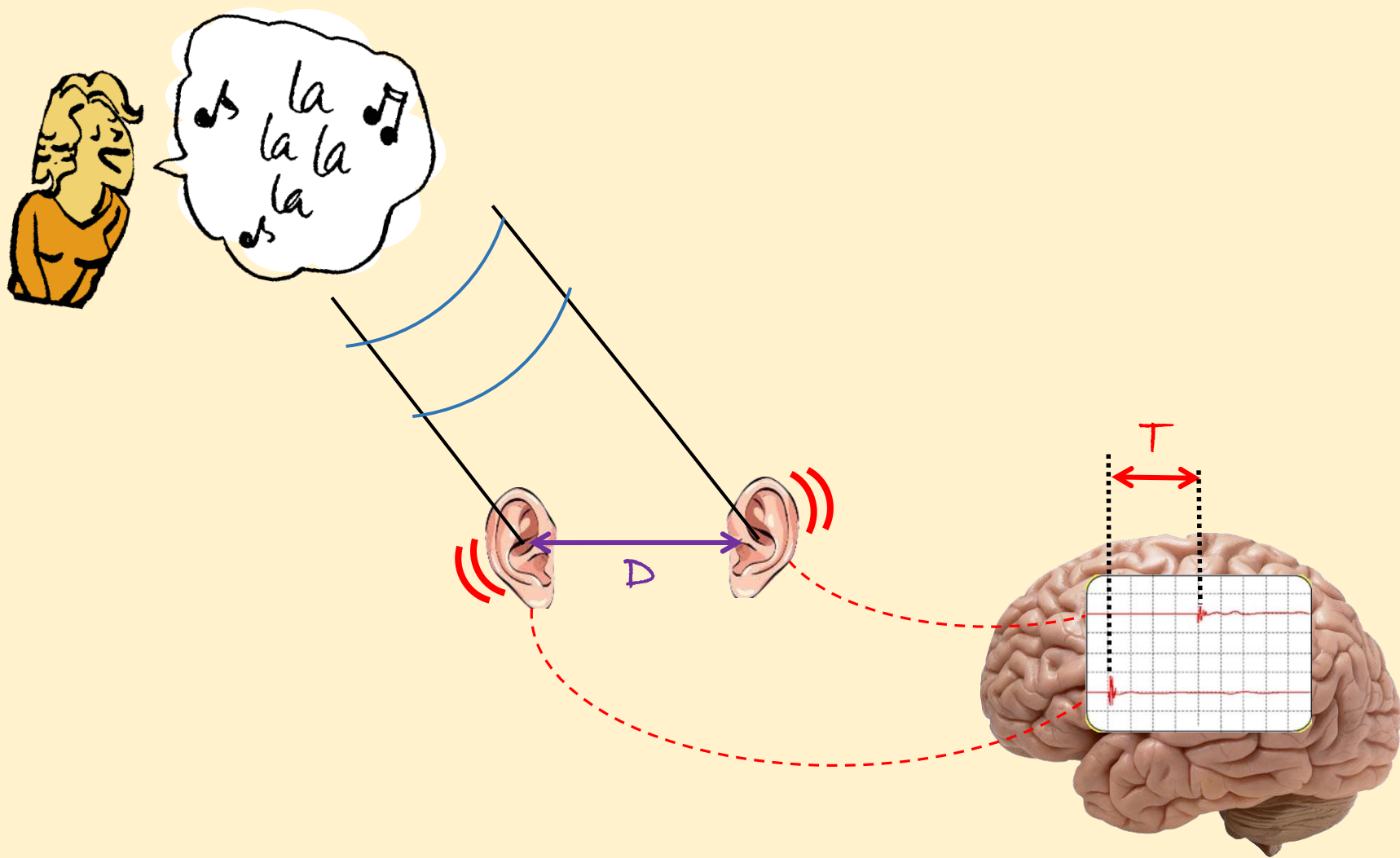


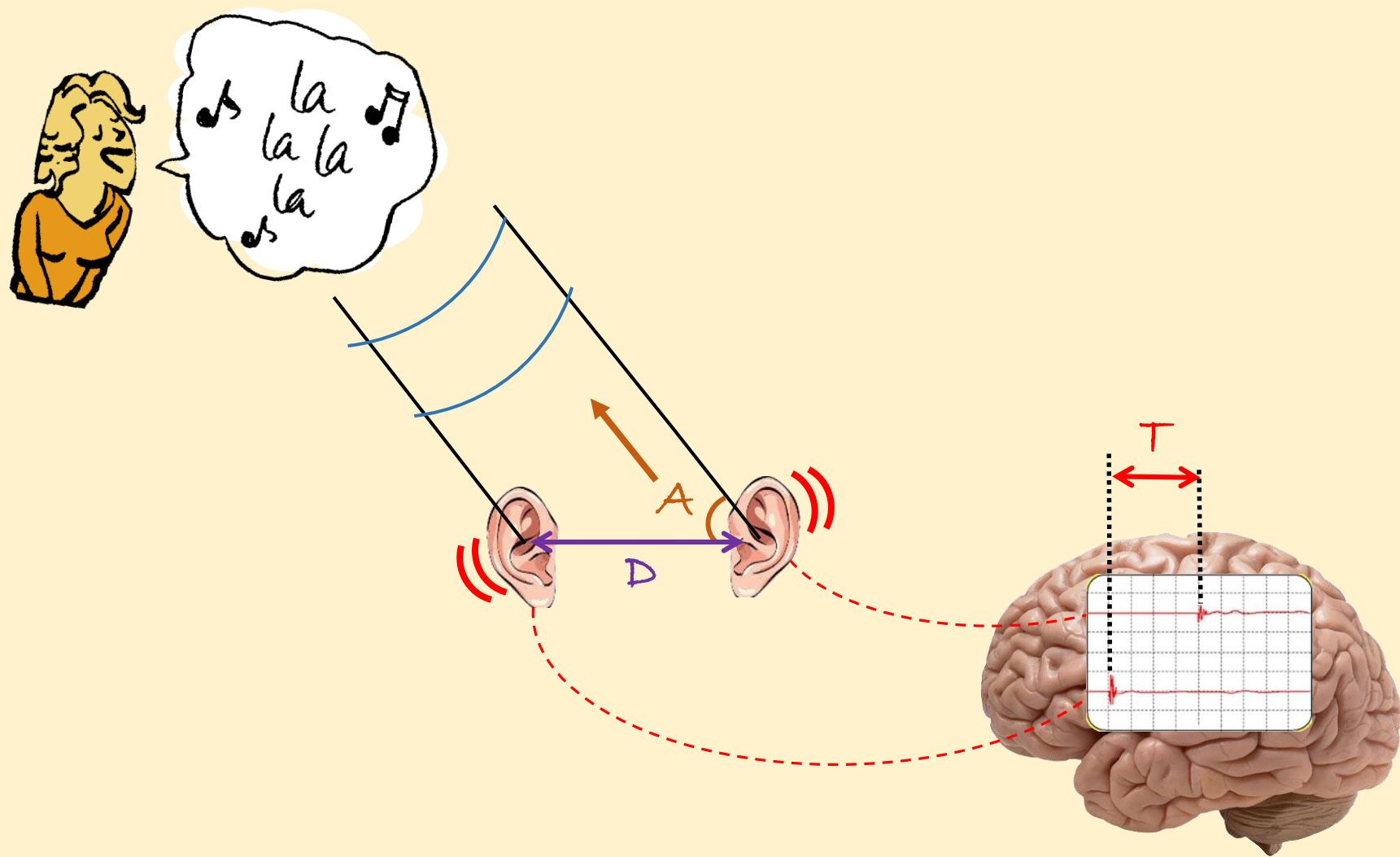


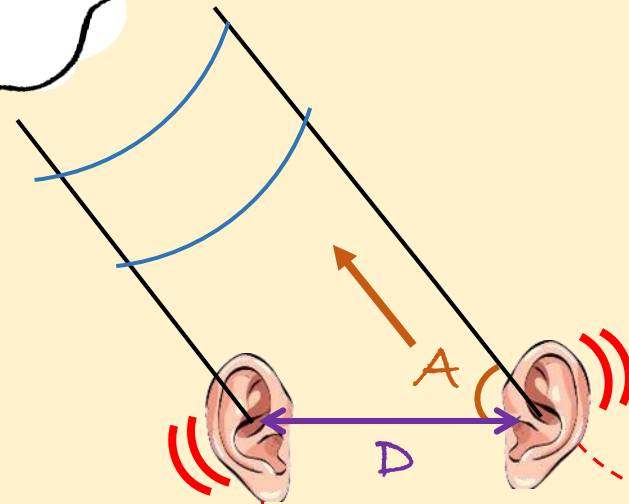




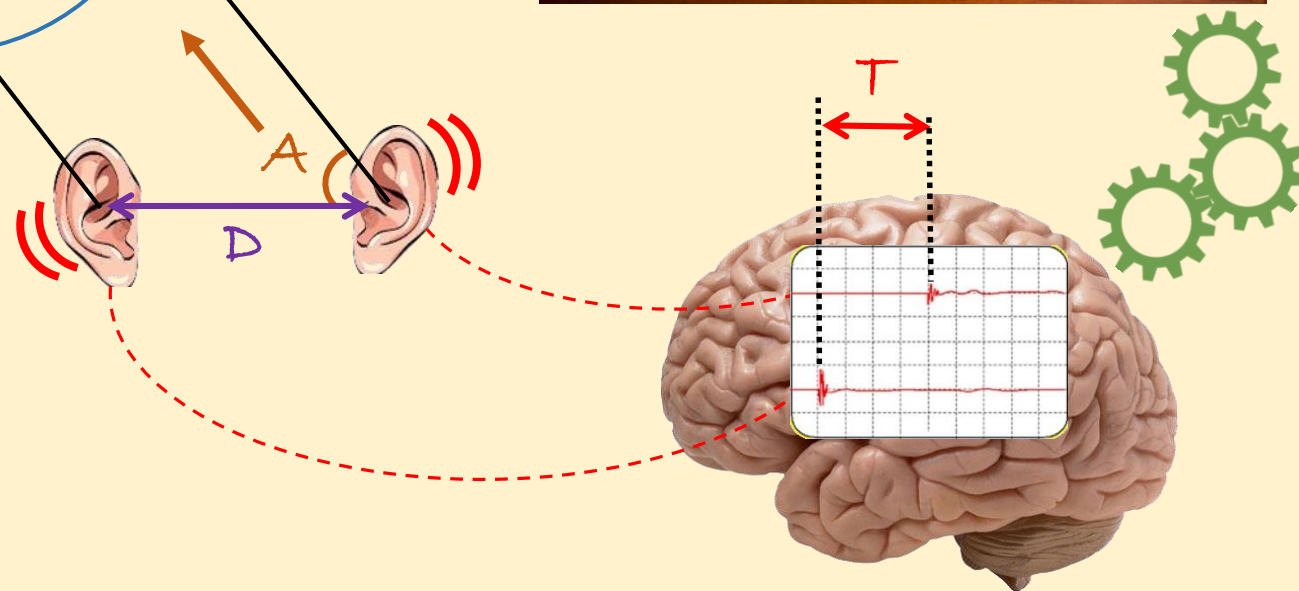






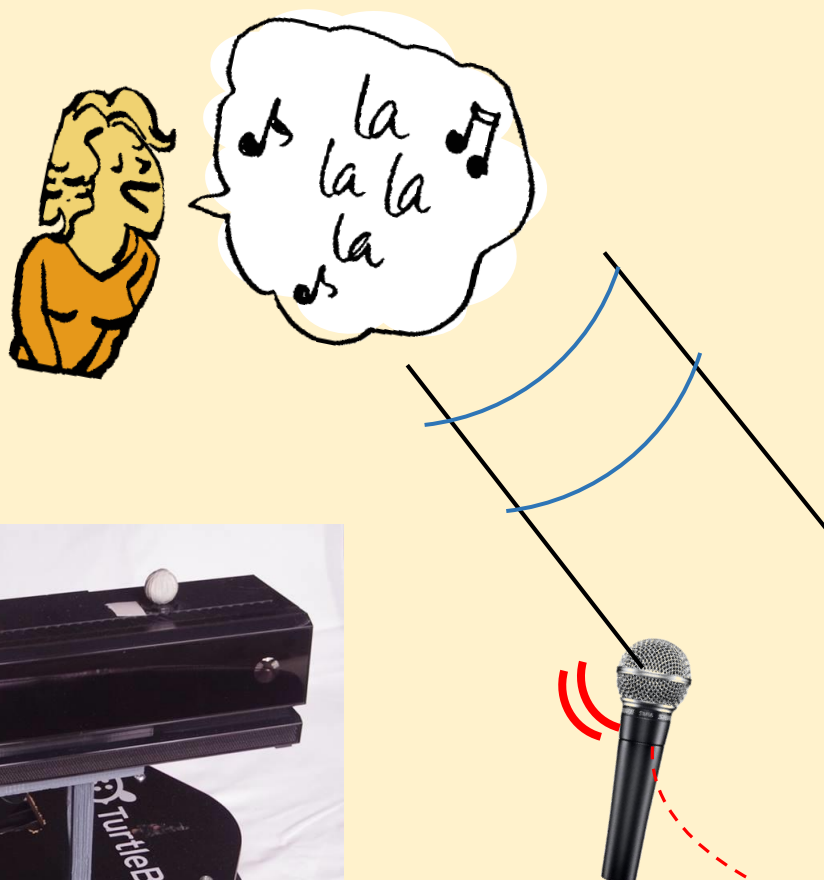


$$A = \cos^{-1} \left( \frac{\text{Speed}}{T \times D} \right)$$

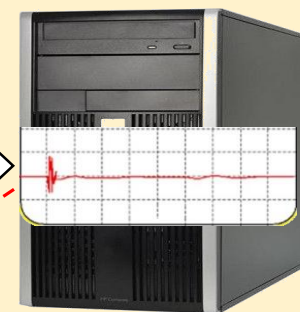


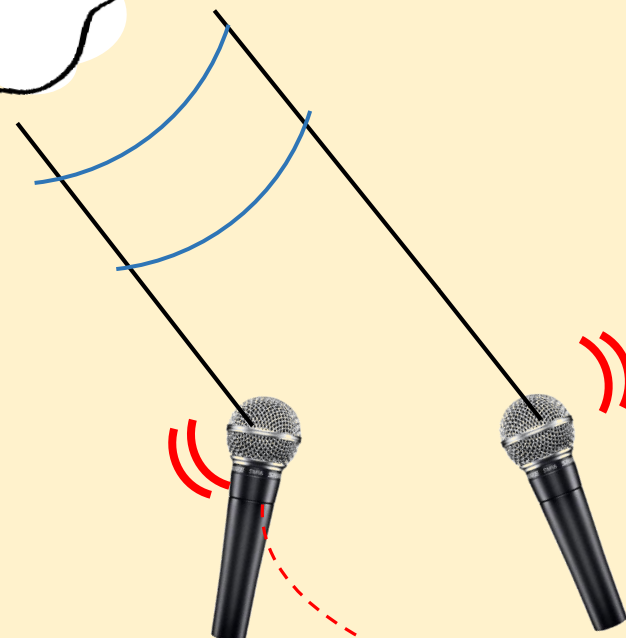




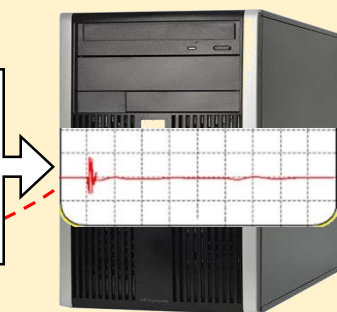


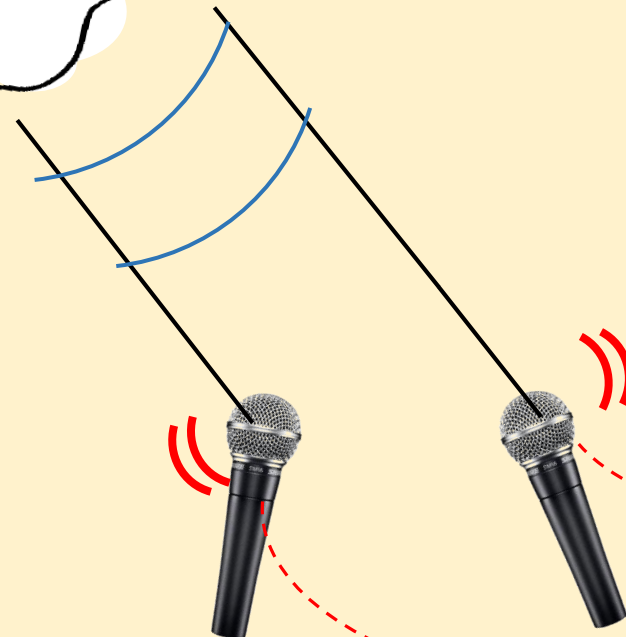
011101010  
010010100  
110100101  
0001001...



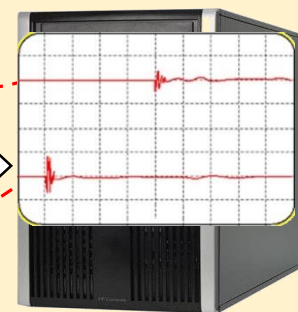


011101010  
010010100  
110100101  
0001001...





011101010  
010010100  
110100101  
0001001...

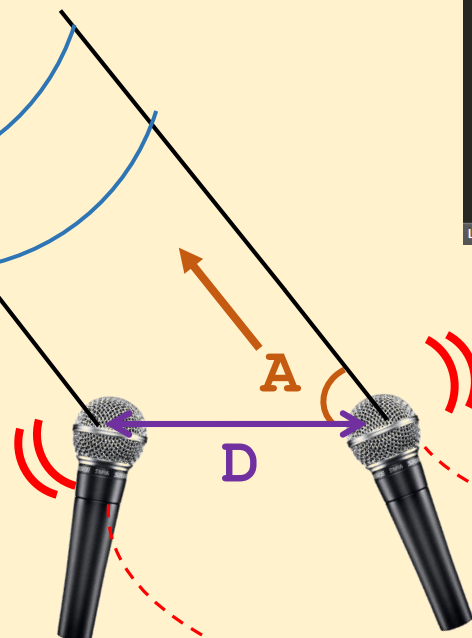




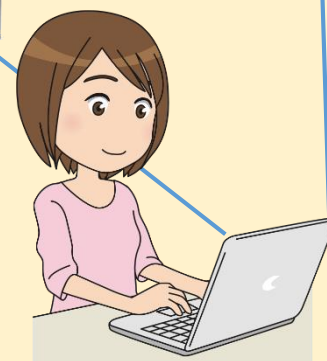
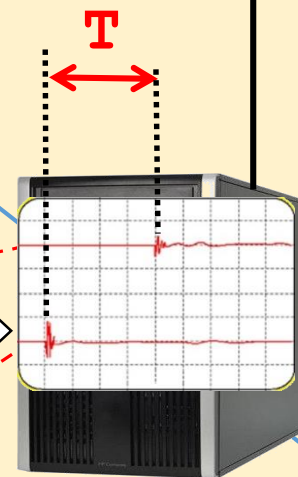
```
base64.cc
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
```

$$A = \cos^{-1} \left( \frac{\text{speed}}{T \times D} \right)$$

Line 31, Column 52      Spaces: 4      C++



011101010  
010010100  
110100101  
0001001...

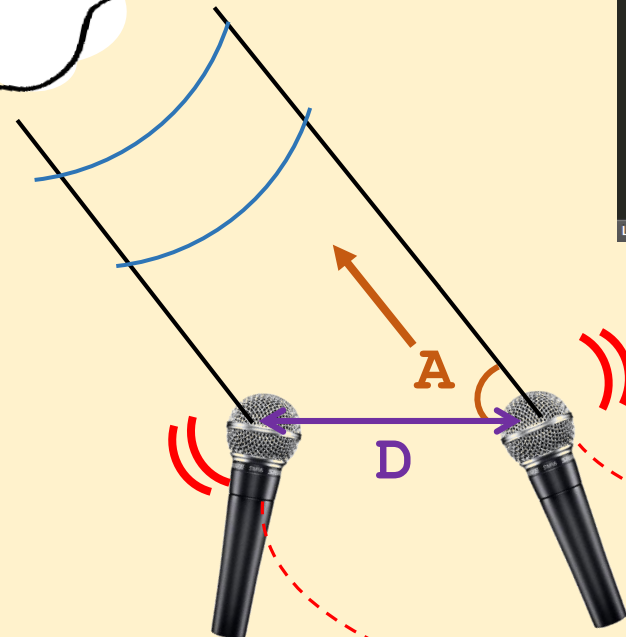




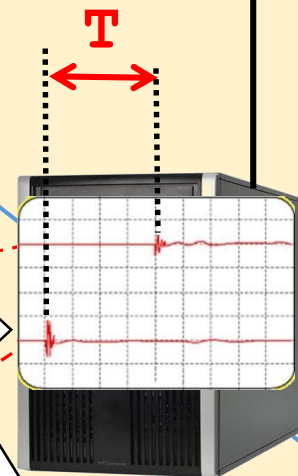
```
base64.cc
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
```

$$A = \cos^{-1} \left( \frac{\text{speed}}{T \times D} \right)$$

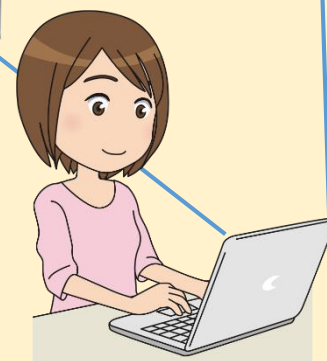
Line 31, Column 52      Spaces: 4      C++



011101010  
010010100  
110100101  
0001001...



011101001  
010011010  
010100001  
00111001...

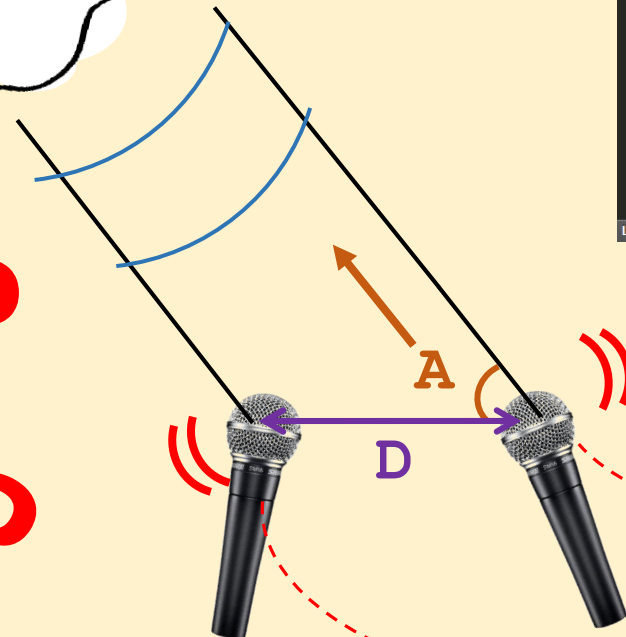
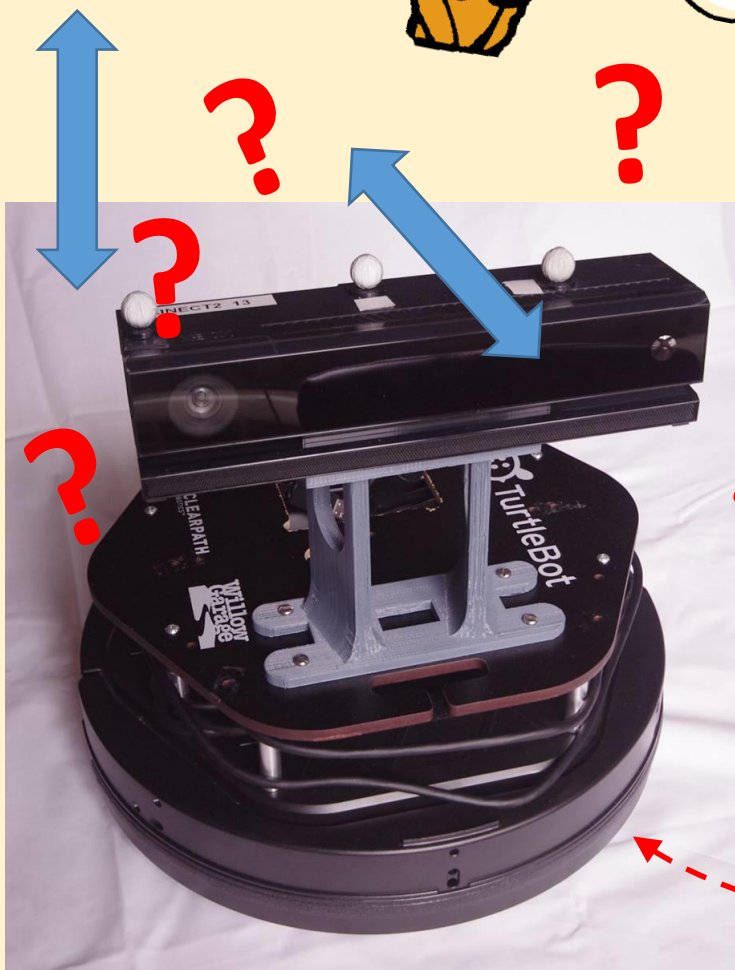




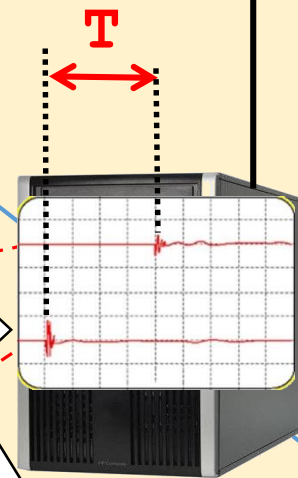
```
base64.cc
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

A = cos^-1 ( speed / (T x D) )

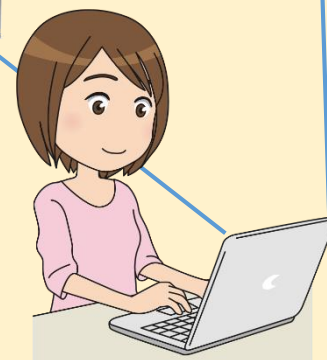
Line 31, Column 52
Spaces: 4
C++
```

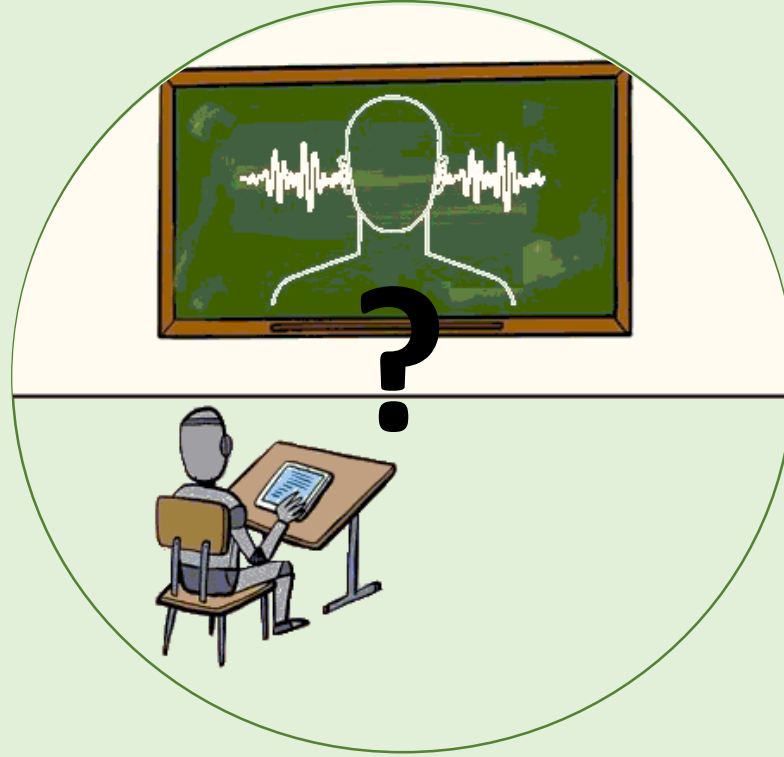


011101010  
010010100  
110100101  
0001001...



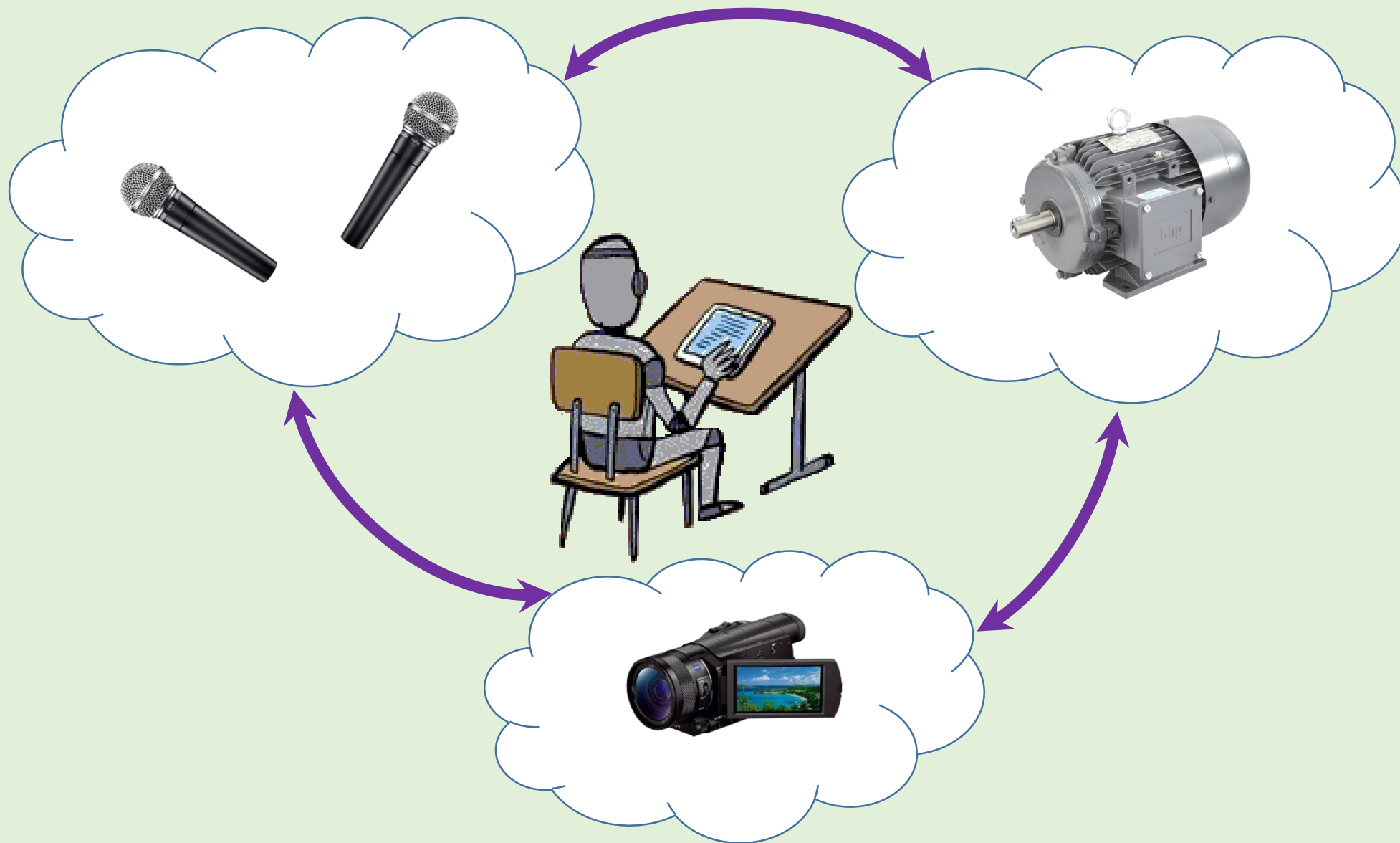
011101001  
010011010  
010100001  
00111001...

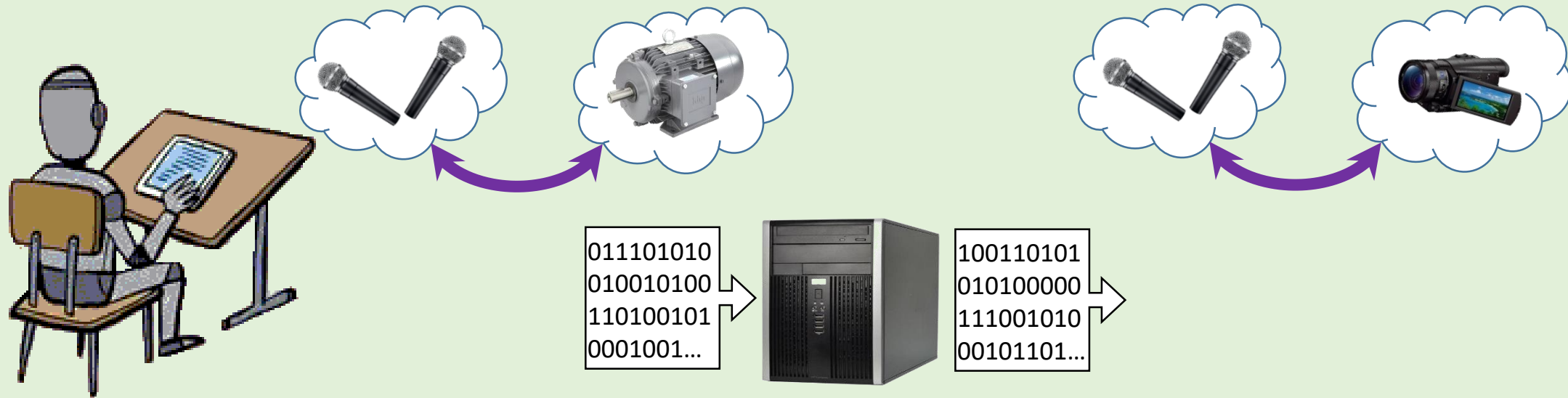
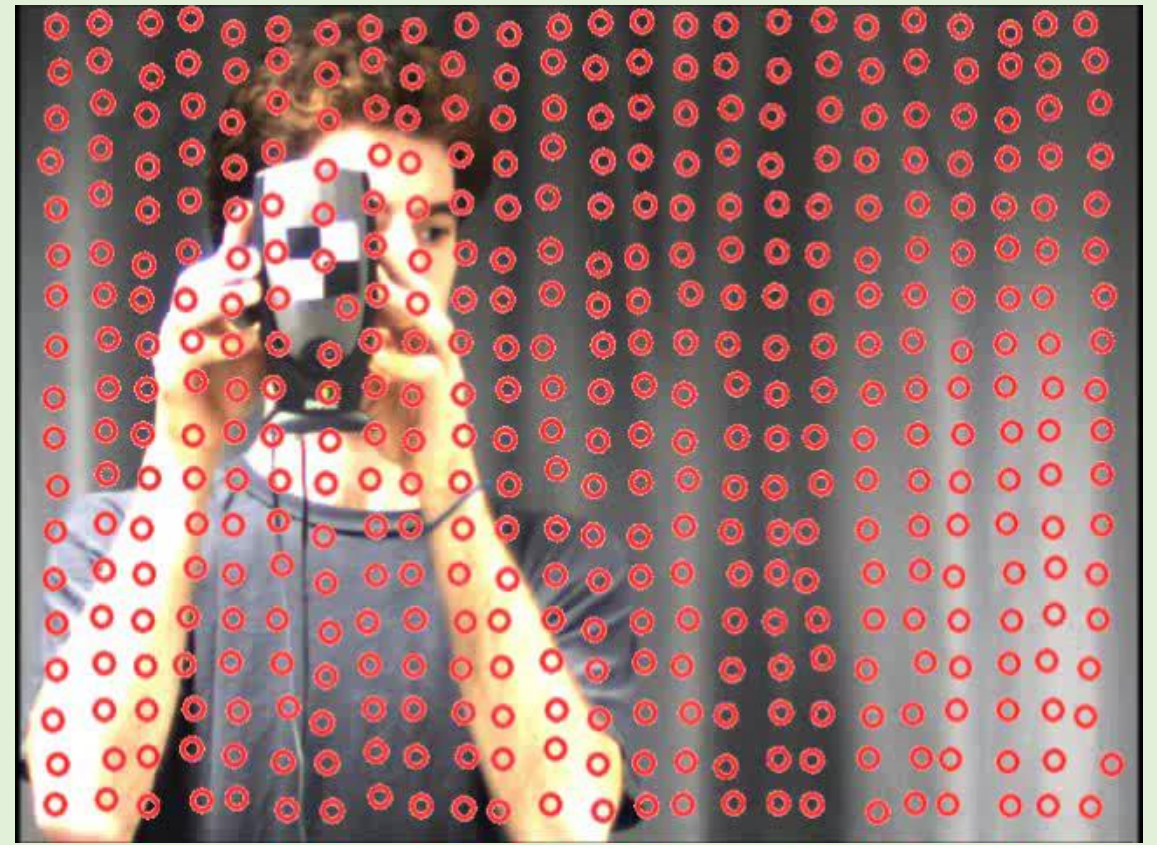


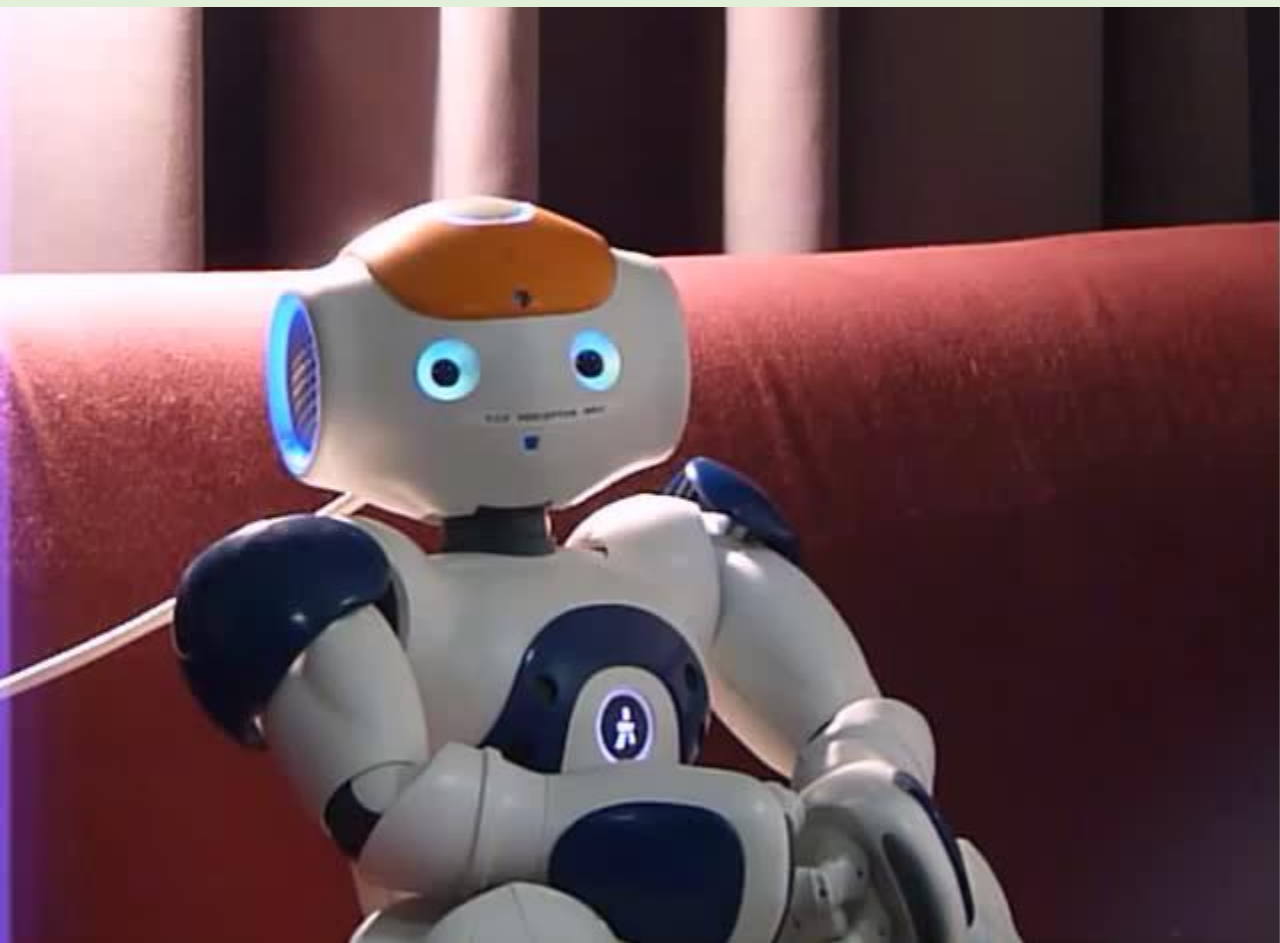


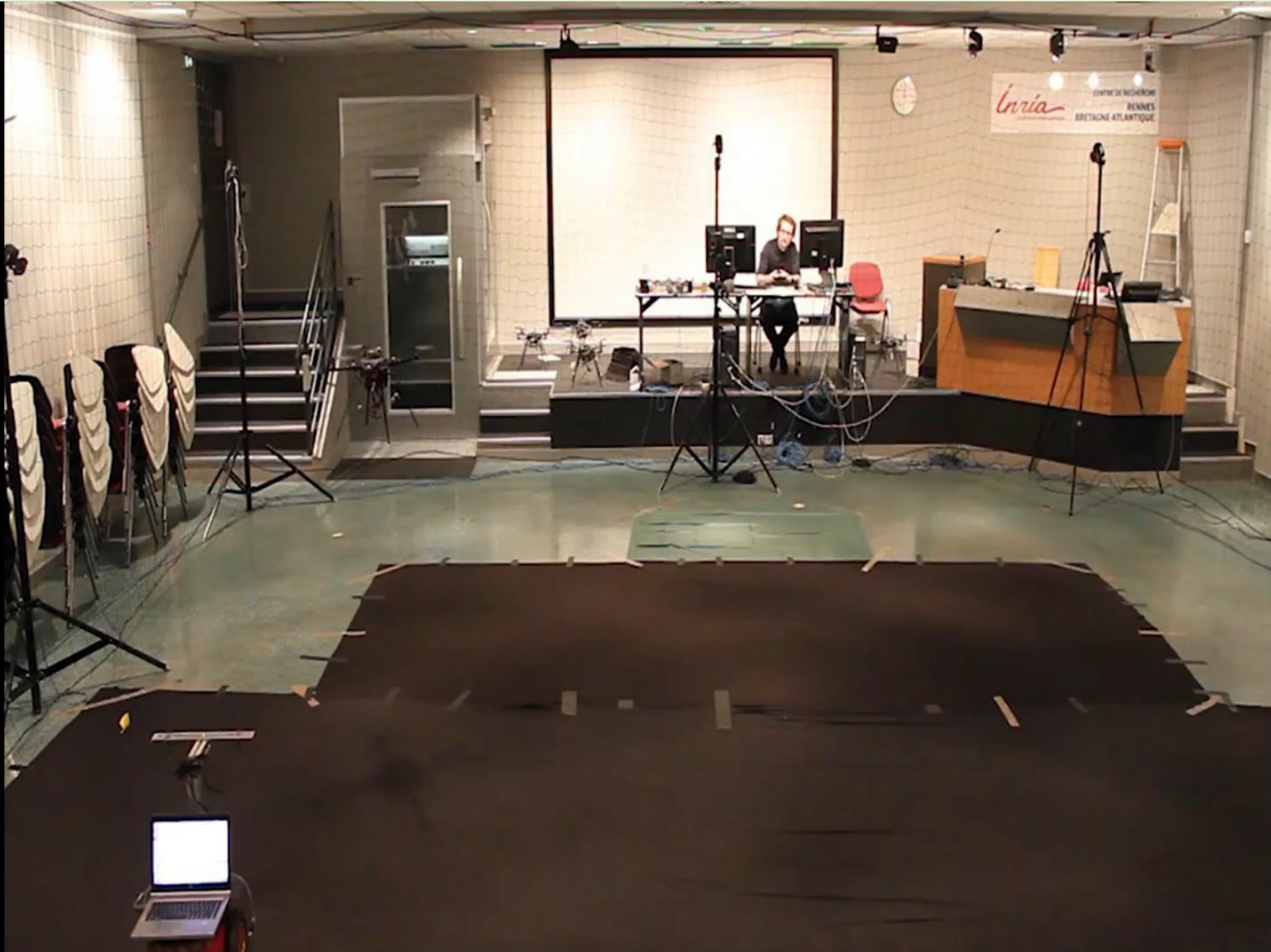












Magassouba, Aly, Nancy Bertin, and François Chaumette. "Sound-based control with two microphones. » IROS 2015

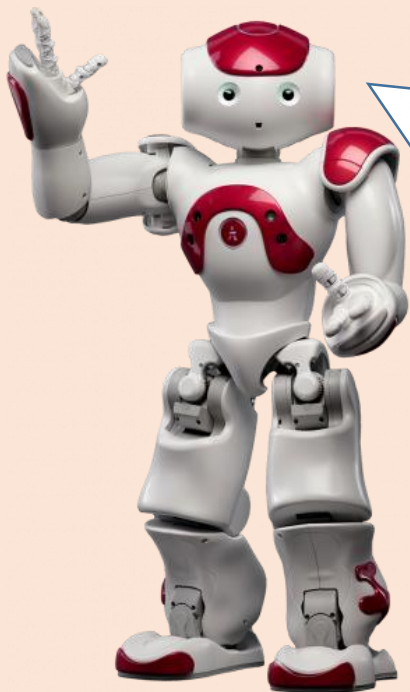


# JOURNÉE SCIENCE & MUSIQUE

Préparation de l'atelier

Game of Drones : le son contre-attaque !





Хвала!  
Hvala!  
Thank you!  
**Merci!**

