

# Пристрій проти гавкання

Бураков Роман, Кірік Анна

# Опис та мотивація

Опис: настільний пристрій, що розпізнає коли собака гавкає, та програє одну з наперед-записаних команд.

Мотивація: допомогти собаці з тривогою розлуки, роблячи вигляд що вона не одна в кімнаті.

# Детекція гавкання

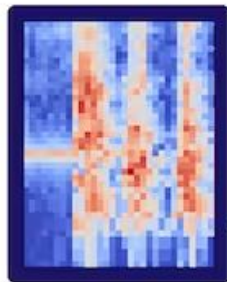
## Фільтрація по гучності (амплітуді)

- Дуже швидко в імplementації
- Ловить багато false-positives
- Складно встановити трешхолд

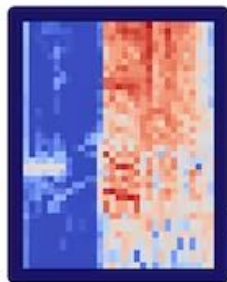
## Keyword detection

- Потребує розмітки даних
- Все ще не складно, враховуючи кількість AutoML сервісів
- Потребує додатковий модуль для обчислень
- Працює дуже точно

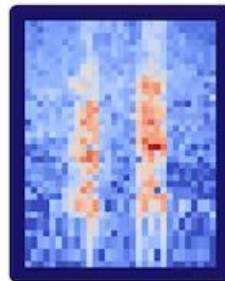
# Детекція гавкання



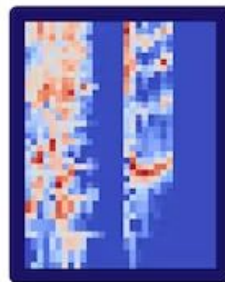
Гавкання



Шум



Гавкання



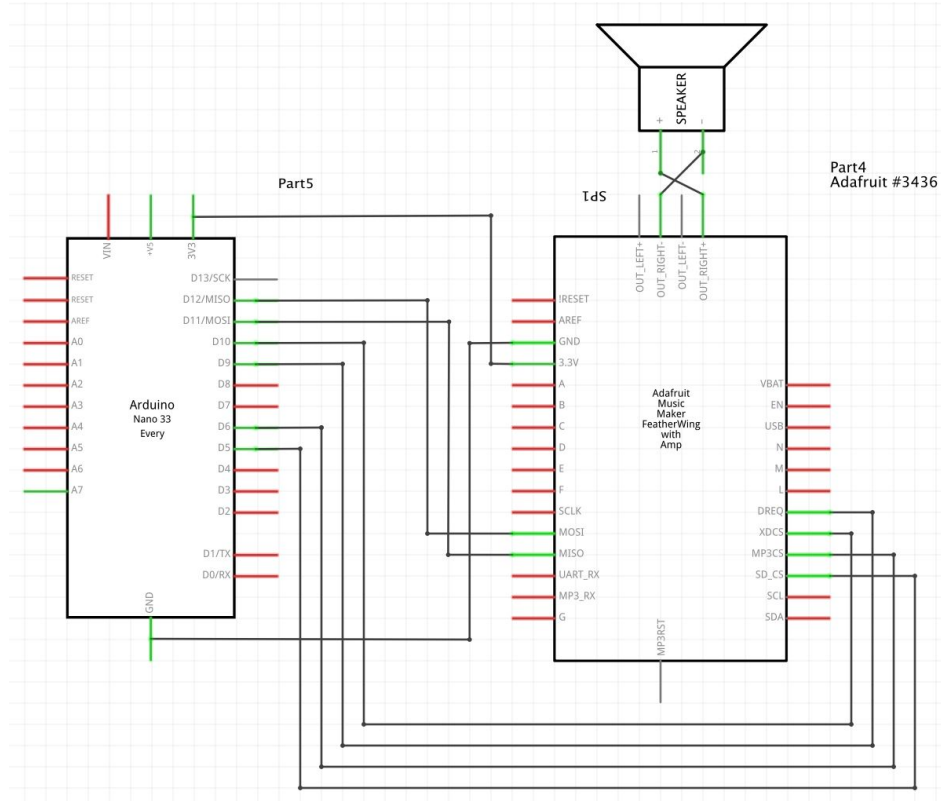
Шум



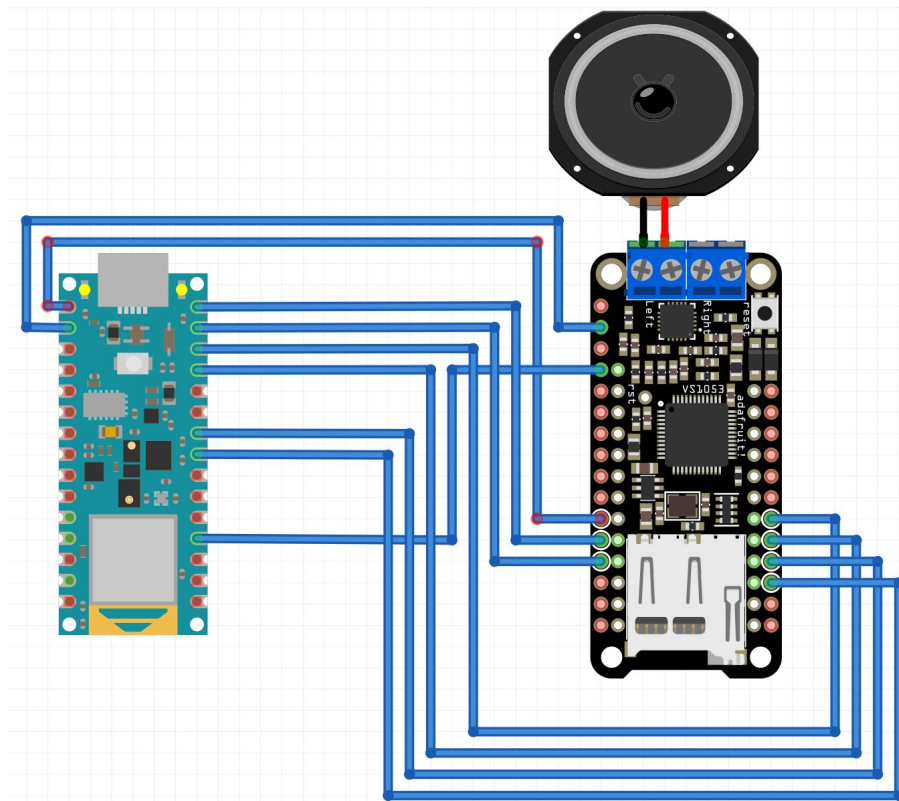
# Компоненти



# Cхема



## Схема (без пінів)



# Код - ініціалізація ([гітхаб](#))

```
1 // #include <PDM.h>
2 #include <bark_detection_inferencing.h>
3 #include <SPI.h>
4 #include <SD.h>
5 #include <Adafruit_VS1053.h>
6 #include <SoftwareSerial.h>
7
8 #define VS1053_RESET -1
9 #define VS1053_CS 6 // VS1053 chip select pin (output)
10 #define VS1053_DCS 10 // VS1053 Data/command select pin (output)
11 #define CARDCS 5 // Card chip select pin
12 #define VS1053_DREQ 9 // VS1053 Data request, ideally an Interrupt pin
13 #define EI_CLASSIFIER_SLICES_PER_MODEL_WINDOW 2
14
15 // Struct to hold inferencing data
16 typedef struct {
17     signed short *buffers[2]; // Buffers to hold audio data
18     unsigned char buf_select; // Index for the current buffer
19     unsigned char buf_ready; // Flag to indicate when a buffer is ready for processing
20     unsigned int buf_count; // Count of samples in the current buffer
21     unsigned int n_samples; // Total number of samples in each buffer
22 } inference_t;
23
24 // Declare variables
25 static inference_t inference;
26 static bool record_ready = false;
27 static signed short *sampleBuffer;
28 static int print_results = -(EI_CLASSIFIER_SLICES_PER_MODEL_WINDOW);
29 Adafruit_VS1053_FilePlayer musicPlayer = Adafruit_VS1053_FilePlayer(VS1053_RESET, VS1053_CS, VS1053_DCS, VS1053_DREQ, CARDCS);
30 const int numTracks = 2;
31
```



# Код - основна частина ([гітхаб](#))

```
33 void setup()
34 {
35     // Initialize serial communication and inferencing
36     Serial.begin(9600);
37     run_classifier_init();
38     // Initialize music player and SD card
39     musicPlayer.begin();
40     SD.begin(CARDCS);
41     musicPlayer.useInterrupt(VS1053_FILEPLAYER_PIN_INT);
42     musicPlayer.setVolume(10, 10);
43     // Start recording audio for inferencing
44     microphone_inference_start(EI_CLASSIFIER_SLICE_SIZE);
45 }
46 void loop()
47 {
48     // Record audio for inferencing
49     bool m = microphone_inference_record();
50     // Set up signal data for inferencing
51     signal_t signal;
52     signal.total_length = EI_CLASSIFIER_SLICE_SIZE;
53     signal.get_data = &microphone_audio_signal_get_data;
54     ei_impulse_result_t result = {0};
55     // Run inferencing on the recorded audio
56     run_classifier_continuous(&signal, &result, false);
57     // If enough slices have been processed, check the results and play a track if necessary
58     if (++print_results >= (EI_CLASSIFIER_SLICES_PER_MODEL_WINDOW)) {
59         if (result.classification[0].value >= 0.7) {
60             int randomTrack = round(random(1, numOfTracks));
61             musicPlayer.playFullFile(String("/") + String(randomTrack) + ".mp3").c_str();
62             delay(10000);
63         }
64         print_results = 0;
65     }
66 }
```

# Висновки

Що вийшло:

- Навчилися шукати потрібні компоненти та читати документацію
- Навчилися проєктувати схеми ардуіно
- Дізналися про дуже швидкий та простий AutoML
- Збір даних

Що не вийшло:

- Реалізація фактичної фізичної системи
- Тестування на собаці

Дякуємо за увагу!