«Звуковая локация»

Проект Шамплетова Никиты

Постановка задачи

Цель: Разработка системы, позволяющей определять положение источник звуковых сигналов.

Требования:

1. Устройство должно анализировать сигналы с нескольких микрофонов (рекомендуется использовать корреляционный анализ) и по задержкам между сигналами определять положение источника звука.
2. Устройство должно корректно определять хотя бы направление на источник.

(Дополнительным плюсом будет определение расстояния до источника)

Математическая модель

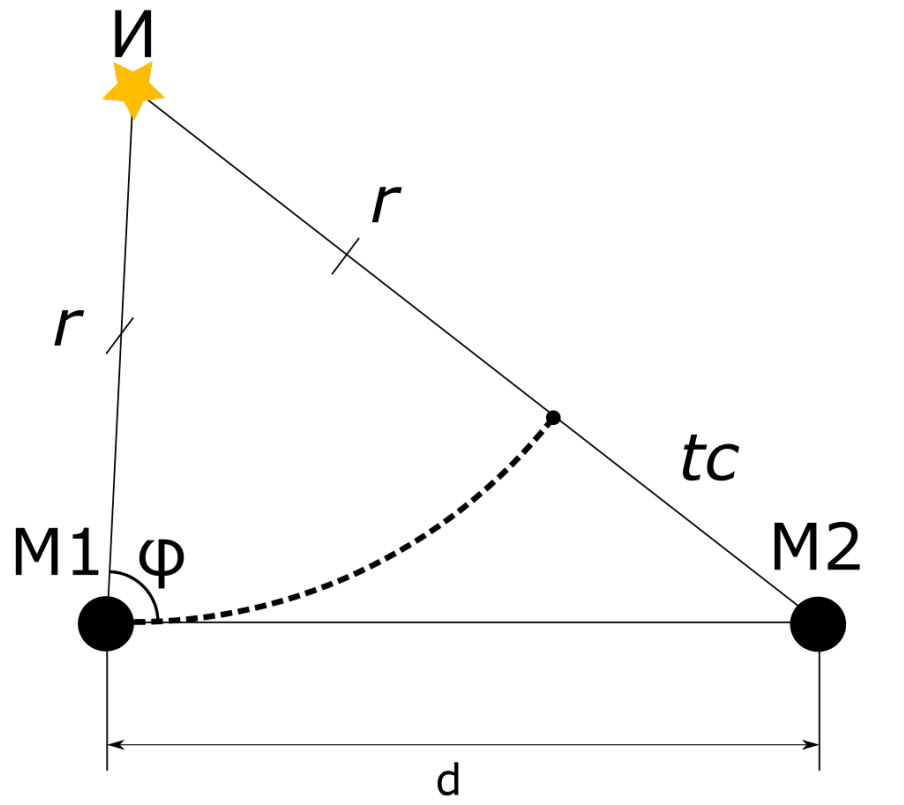


Рис.1. Источник и два микрофона

И – источник

M1 – микрофон №1

M2 – микрофон №2

t – задержка по времени

c – скорость звука

Запишем теорему косинусов для ΔИM1M2:

$$(r + tc)^2 = r^2 + d^2 - 2\cos(\phi)rd$$

$$2rtc + (tc)^2 = d^2 - 2\cos(\phi)rd$$

$$2r(tc + d\cos(\phi)) = d^2 - (tc)^2$$

$$r(\phi) = \frac{d^2 - (tc)^2}{2(tc + d\cos(\phi))}$$

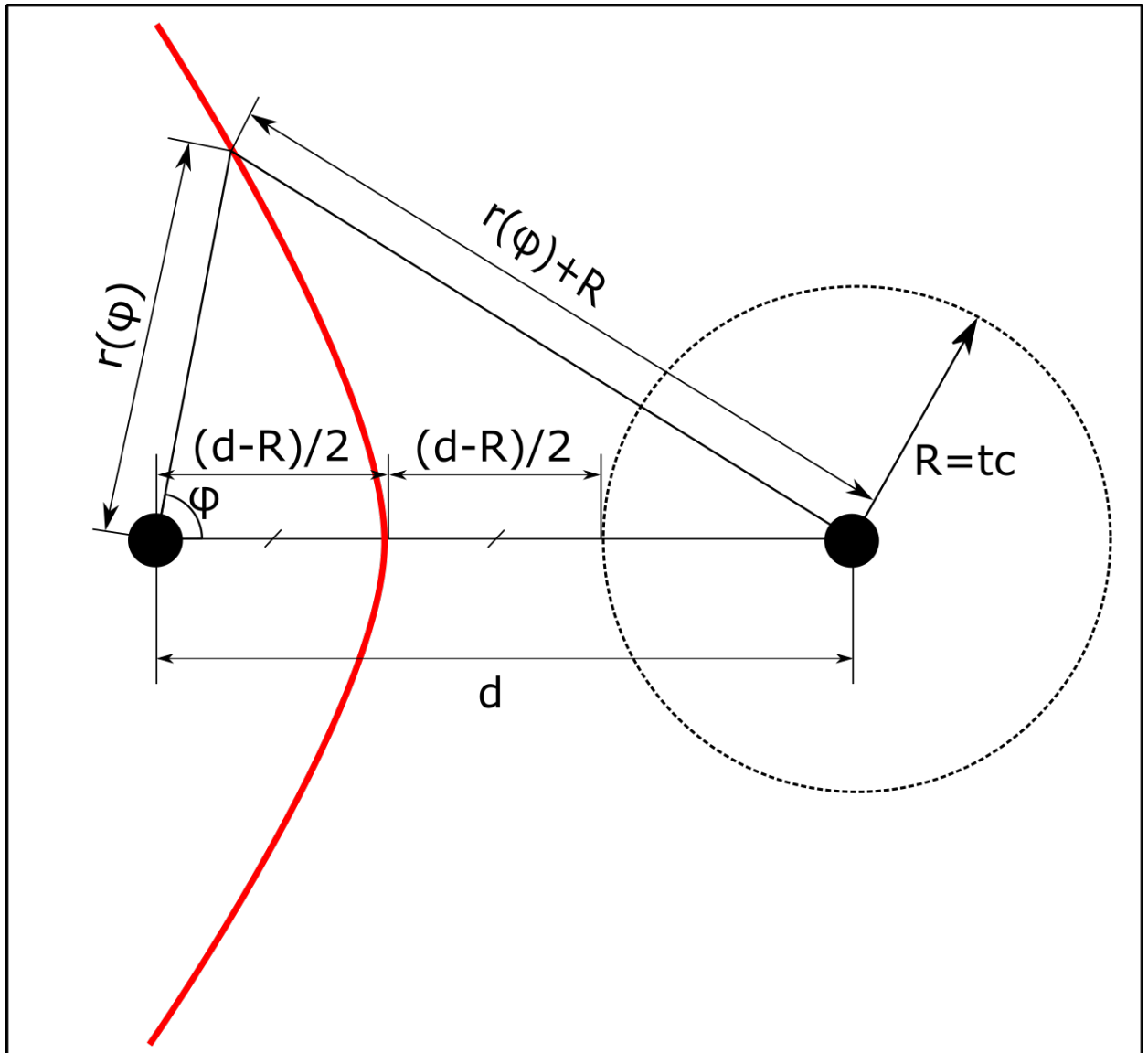


Рис.2. Кривая источников
и два микрофона

$$r(\phi) = \frac{d^2 - (tc)^2}{2(tc + dcos\phi)} = \frac{d}{2}\frac{1 - n^2}{n + cos\phi}$$

$$n \equiv tc/d$$

Работа в 3D

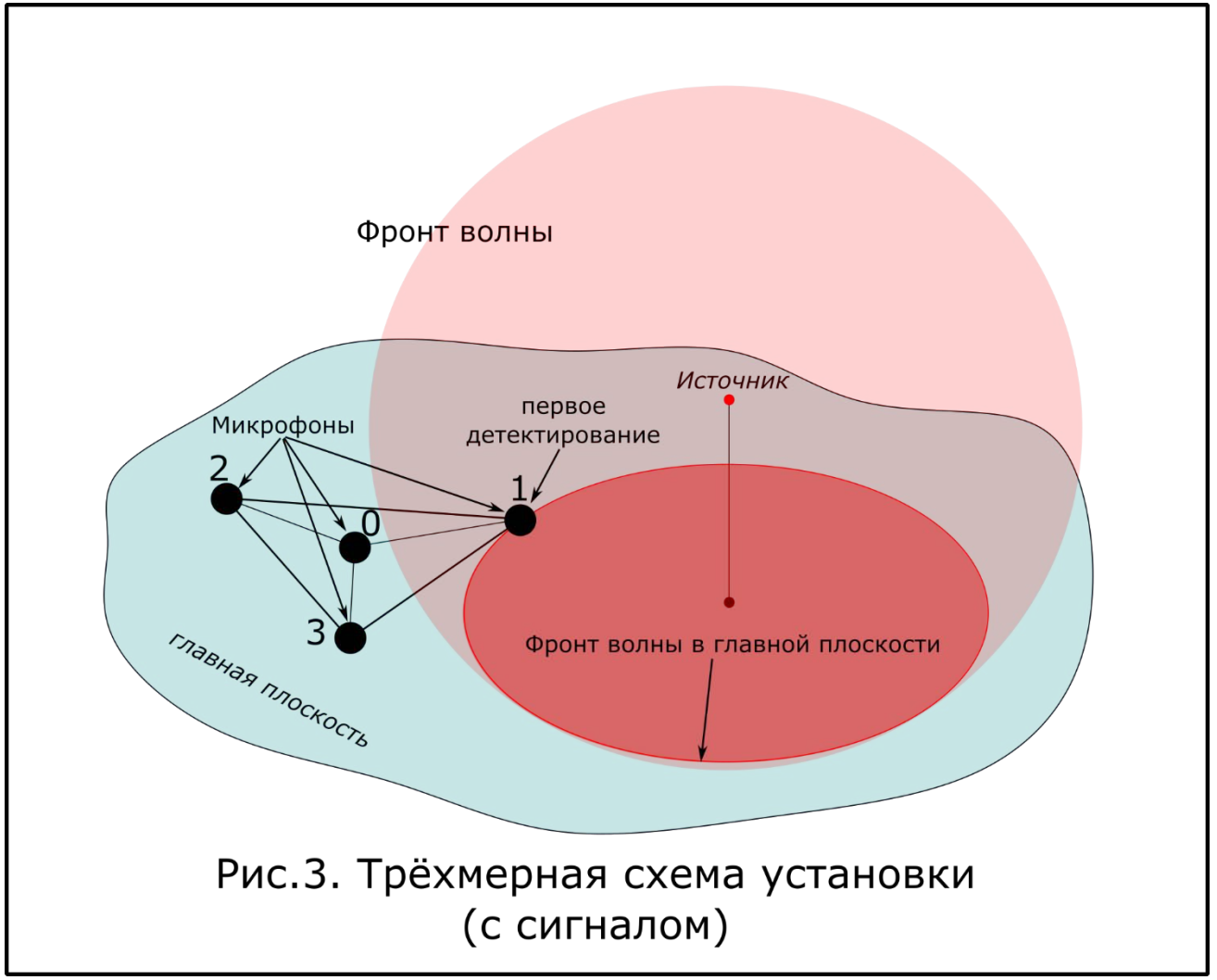


Рис.3. Трёхмерная схема установки
(с сигналом)

Было решено использовать расположение микрофонов «треугольником» (равносторонним) с размещением одного из этих микрофонов в центре.

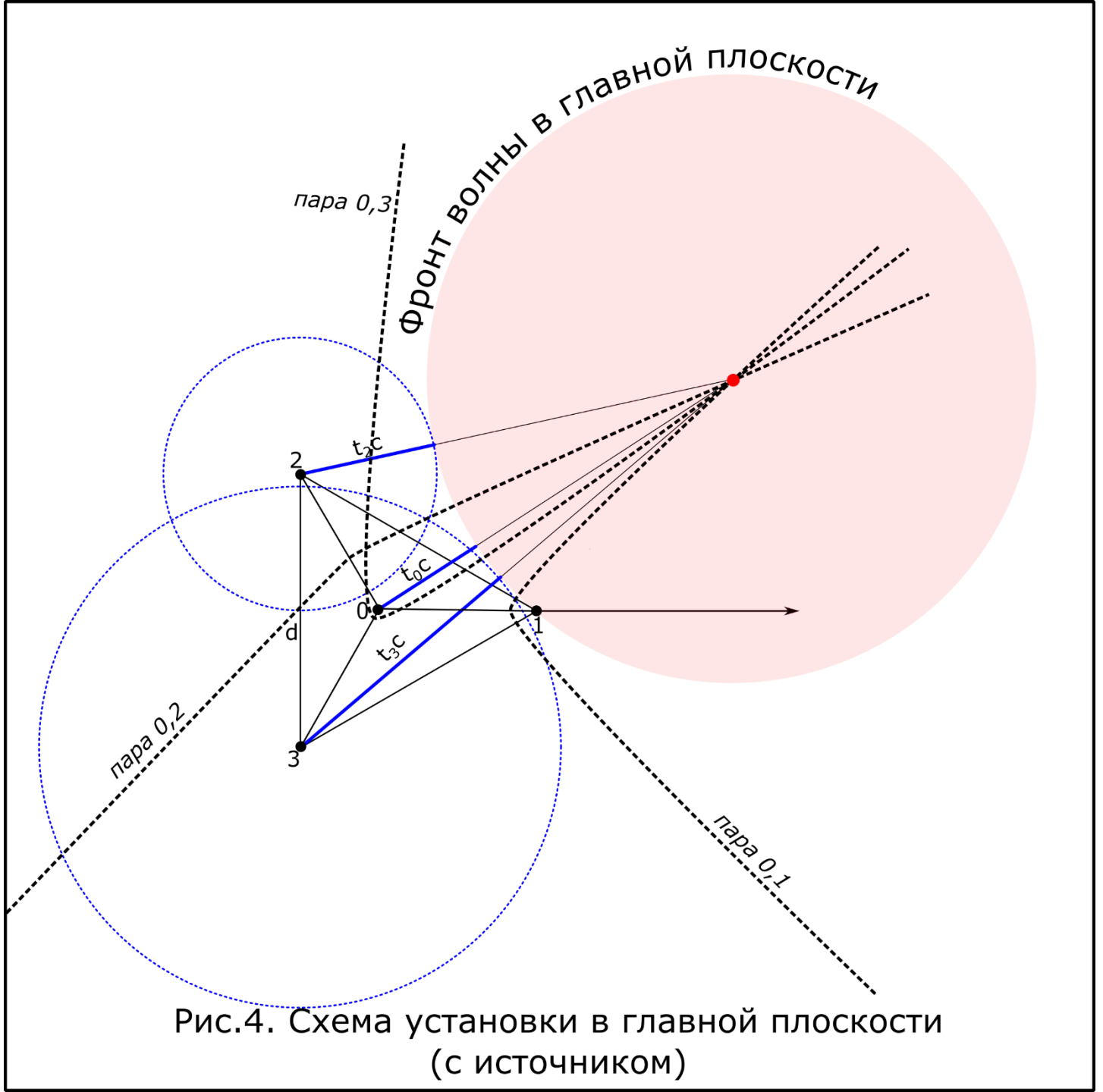


Рис.4. Схема установки в главной плоскости (с источником)

Математика для случая 3D

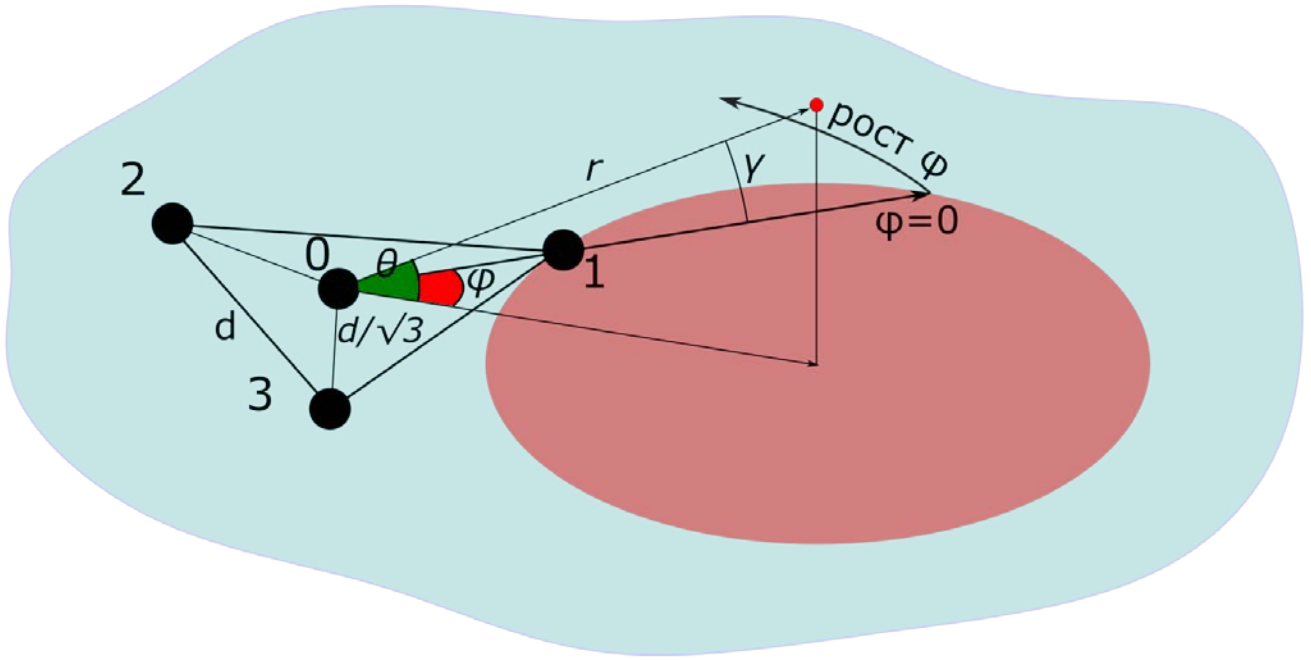


Рис.5. К определению положения источника

$$r(\phi) = \frac{(d/\sqrt{3})}{2} \frac{1-n^2}{n+\cos\gamma}$$

$$n \equiv tc/(d/\sqrt{3})$$

Можно показать, что

$$\cos\gamma = \cos\phi\cos\theta$$

Выражения для расстояний от микрофона №0 до источника

$$r_1(\phi,\theta) = \frac{d}{2\sqrt{3}} \frac{1-n_1^2}{n_1 + \cos\phi\cos\theta}$$

$$r_2(\phi,\theta) = \frac{d}{2\sqrt{3}} \frac{1-n_2^2}{n_2 + \cos(\phi - 2\pi/3)\cos\theta}$$

$$r_3(\phi,\theta) = \frac{d}{2\sqrt{3}} \frac{1-n_3^2}{n_3 + \cos(\phi + 2\pi/3)\cos\theta}$$

Из этих формул и получаются все три интересующие величины: расстояние и два угла.

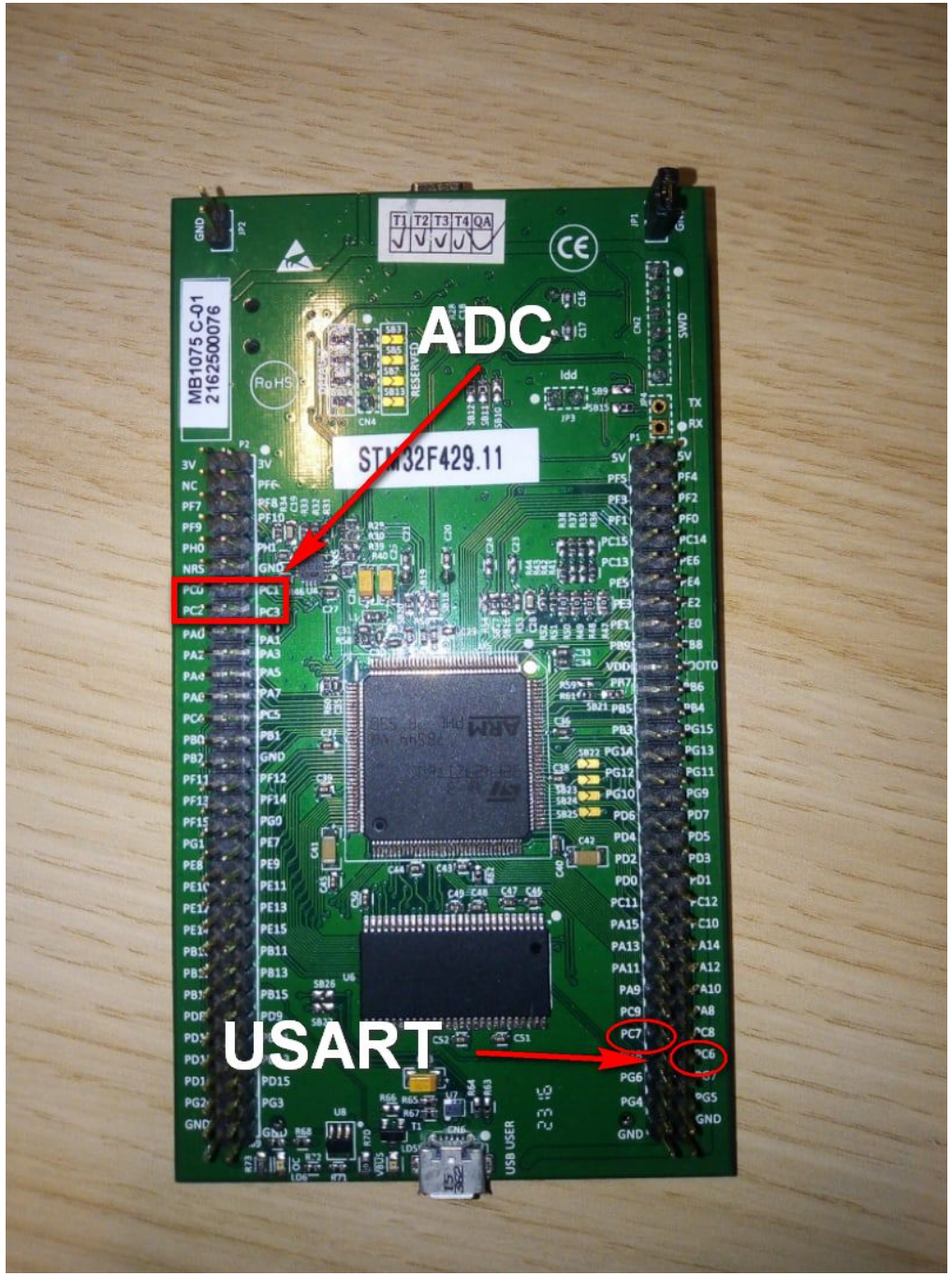
Реализация. Прошивка

Использующиеся блоки:

- GPIO
- TIM (TIM3)
- ADC (ADC1)
- DMA (DMA2)
- USART (USART6)

GPIO:

PC0,1,2,3 – ADC(CH10,11,12,13)
PC6 – USART Tx
PC7 – USART Rx



ADC и DMA

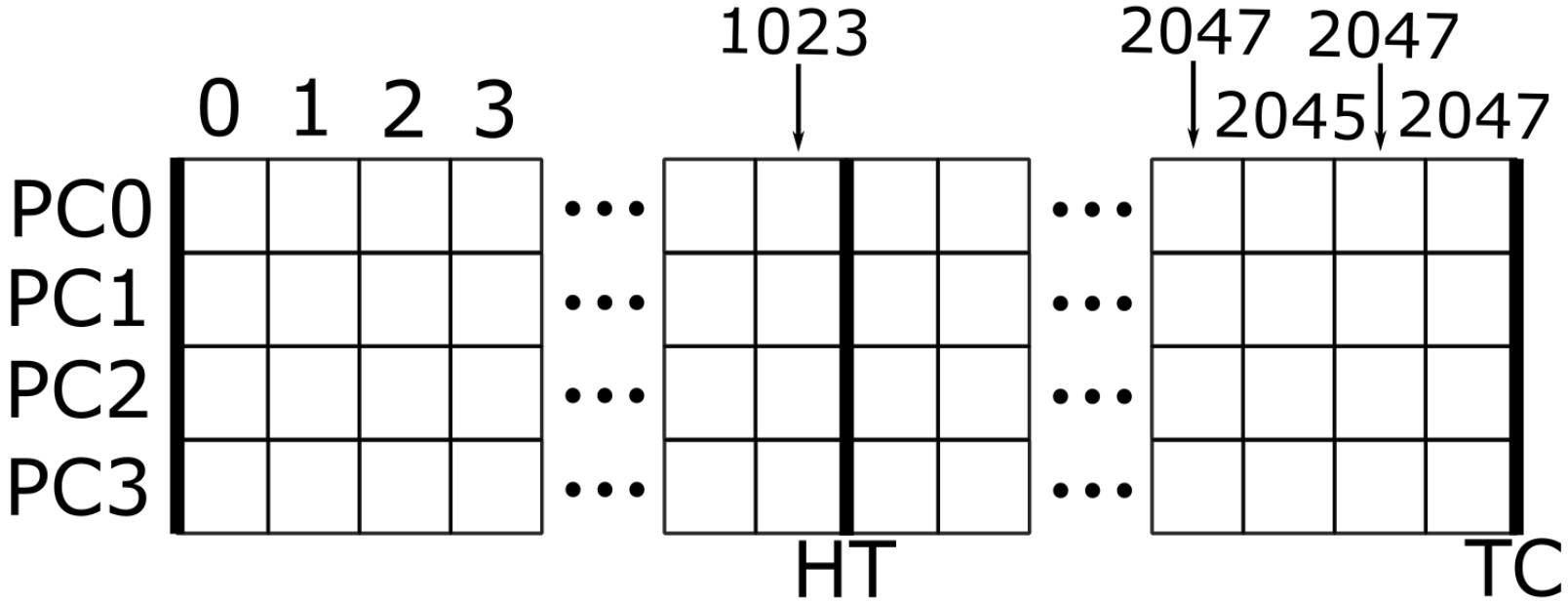


Рис.6. Массив данных, заполняемый DMA

ADC работает в режиме *scan:* после каждого сигнала таймера (trigger) блок выполняет 4 считывания, а именно с PC0, PC1, PC2 и PC3. Параллельно DMA пересылает получаемые значения в буфер на 4×2048.

Обработка данных осуществляется по частям: когда DMA пересылает половину данных, в main() начинается обработка готовой части массива, после которой обработка останавливается до тех пор, пока DMA вновь не заполнит половину буфера.

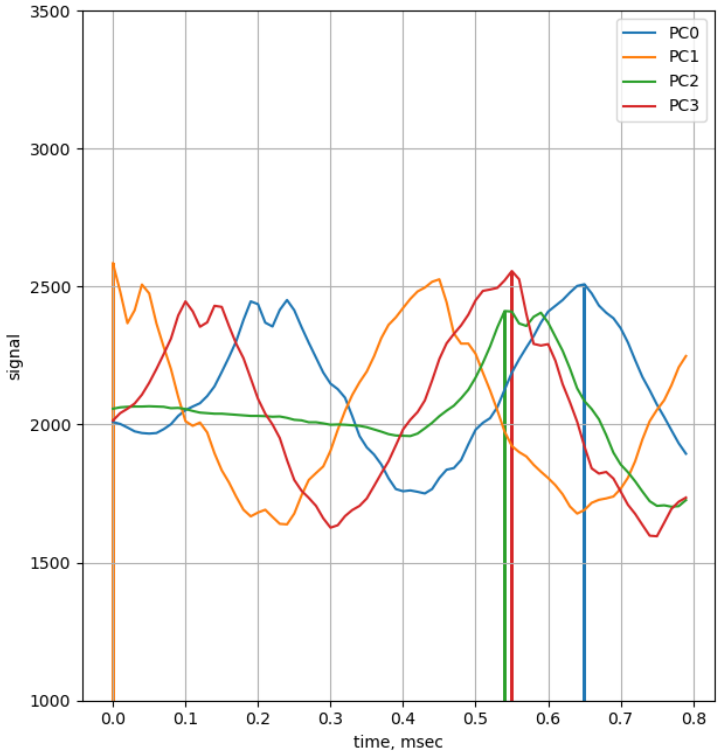
Что отсылает плата?

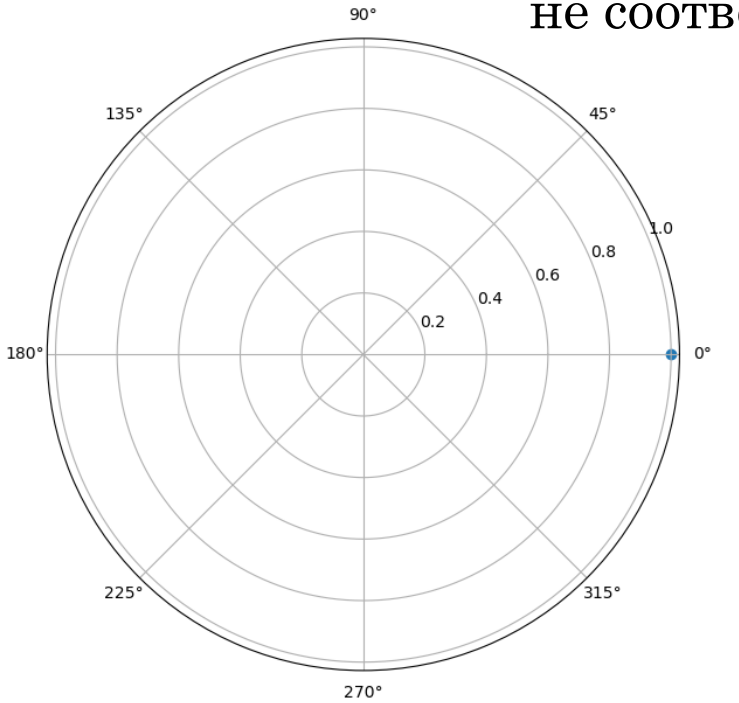
Прежде всего ARR, PSC и размер отсылаемого массива данных.

Было сделано *предположение* о том, что максимумы сигнала на разных каналах соответствуют одной и той же фазе. Поэтому плата *ищет в сигнале максимумы*, определяет их положения в массиве и отсылает эти положения на ПК. В придачу плата отправляет ещё и ту область буфера, в которой были обнаружены пики.

Замечу, что для ускорения отправки применяется перевод в 16-тиричную систему, это позволяет сократить число передаваемых символов на 25%!

Трудности





Максимумы на разных каналах не соответствуют одной фазе!

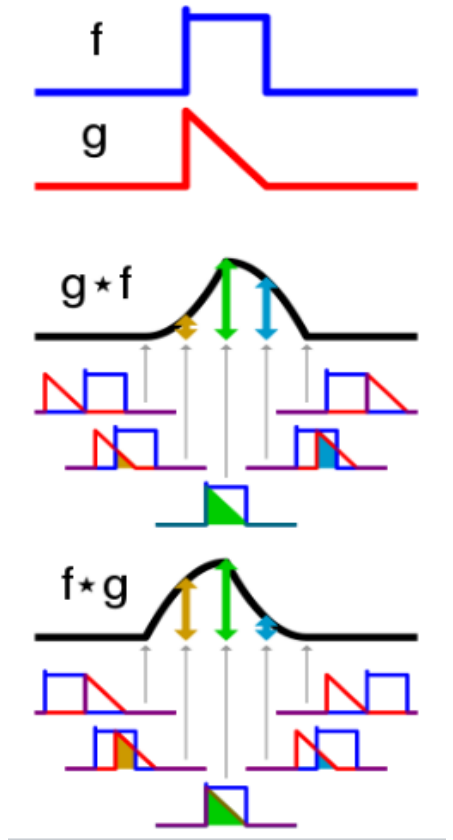
Использование корреляции

Для решения проблемы решено было использовать корреляционную функцию из библиотеки NumPy:

numpy.correlate(f,g),

где f и g – сравниваемые массивы.

К сожалению, поиск задержек с помощью корреляции тоже работает не всегда.



C:\WINDOWS\system32\cmd.exe - python SCH.py

```
altitude, degrees: 0
---------- ----------
No warnings. Trying get correlated position...
---------- ----------
Get following n's: -0.51, -0.30, 0.76
radiuses, cm: -222, 31, 2
polar angle, degrees: 50
altitude, degrees: 38
---------- ----------
========== ==========

========== ==========
starting read the port...
signal transmition is complete!
starting read the signal positions...
positions have been successfully transmitted and converted!

Start correlation...
correlation completed!

Start checking operations...
No warnings. Trying get position...
---------- ----------
Get following n's: 0.06, 0.09, 0.24
radiuses, cm: 48, 39, 19
polar angle, degrees: 51
altitude, degrees: 83
---------- ----------
ERROR: Incorrect correlated n!
n: 2.81, 0.15, 0.30
========== ==========
```