Московская предпрофессиональная олимпиада

Заключительный этап по направлению
«Инженерно-конструкторское. Аэрокосмический профиль»

Время выполнения задания – 235 мин.

“Дублирование работающего малого космического аппарата Cubesat 1U”

Заключительный этап проводится формате выполнения практического задания, состоящего из нескольких этапов (3D-моделирование, обработка значений, программирование микроконтроллера, кодирование данных). В рамках данного задания участники олимпиады должны проявить свои междисциплинарные знания в области инженерии космических систем (аппаратные платформы, микроконтроллеры, 3D-моделирование, разработка программного обеспечения, анализ данных).

Задание: в соответствии с требованиями каждого этапа выполнения задания заключительного этапа олимпиады команде участников необходимо разработать прототип собственного малого космического аппарата (МКА) формата Cubesat 1U, оснащённого бортовым компьютером, солнечными панелями, системой автономного питания и батареями, а также бортовыми датчиками температуры и магнитного поля. Формат предоставления показаний датчиков должен соответствовать формату представления данных работающего аппарата, представленного на испытании в виде “чёрного ящика”, отправляющего закодированную информацию через радиопередатчик.

Примечание: каждый этап выполнения задания заключительного этапа олимпиады может выполняться отдельно от других этапов и/или выполняться параллельно с другими этапами.

Этап № 1. Определение формата передаваемых данных.

При помощи микроконтроллера Arduino UNO и модуля радиоприёмника требуется реализовать приём показаний датчиков, отправляемых с радиопередатчика внутри “чёрного ящика”, а также произвести их расшифровку.

Перечень необходимых элементов: Микроконтроллер Arduino, радиоприёмник FM433 Hz (Приложение №4).

В рамках выполнения этапа № 1 команде участников необходимо:

1. Разработать устройство приёма радиосигнала, отправляемого с “чёрного ящика” и реализовать программный код приёма закодированного сообщения и вывода его на экран.

2. Произвести расшифровку полученного сообщения, переведённого в HEX код с помощью кодировки ASCII автоматически или вручную. В случае, если команде не удалось принять сообщение с “чёрного ящика”, его можно взять из текстового файла приложение № 1.

3. Разработать программное обеспечение, способное производить автоматизированную раскодировку сообщения с “чёрного ящика”, состоящего из латинских букв верхнего и нижнего регистра, цифр и символов “=” (равно) и “_” (нижнее подчёркивание).

Алгоритм работы программного обеспечения “чёрного ящика”:

“Чёрный ящик” транслирует сообщения посредством радиопередатчика. Каждое сообщение содержит в себе строку, формата: “**название_отправляемого_параметра=значение_параметра**” (пример: **new_temp=20**), либо текстового сообщения на английском языке с символами “_” (нижнее подчёркивание) вместо пробелов. Длина каждого сообщения перед кодировкой составляет 15 символов, если значимых символов в сообщении было меньше 15, то они дополняются до 15 символами “_” (нижнее подчёркивание) (например:**new_temp=20** перед кодированием будет дополнен до **new_temp=20**____).“Чёрный ящик” передаёт неизменяемый по своей структуре и содержанию набор сообщений, отображающий показания датчиков (показания датчиков одинаковые для каждого сообщения), и информацию о самом ящике (название и идентификационный номер). Сообщения передаются с паузой в 1 секунду, повторная отправка пакета производится спустя 5 секунд после отправки пакета.

Примечание: ASCII (англ. *American standard code for information interchange*) – название таблицы (кодировки, набора), в которой некоторым распространённым печатным и непечатным символам сопоставлены числовые коды. Каждому символу ставится в соответствие восьмиразрядное двоичное число – байт. Поскольку байт можно представить в виде двух шестнадцатеричных цифр (старшие четыре бита и младшие четыре бита), коды символов нередко записывают в шестнадцатеричном представлении для экономии места. Далее пример перевода сообщения из двоичного представления в текст, с использованием шестнадцатеричного представления как промежуточного. Таблица кодирования ASCII в шестнадцатеричной записи (HEX) представлена в приложении № 2.

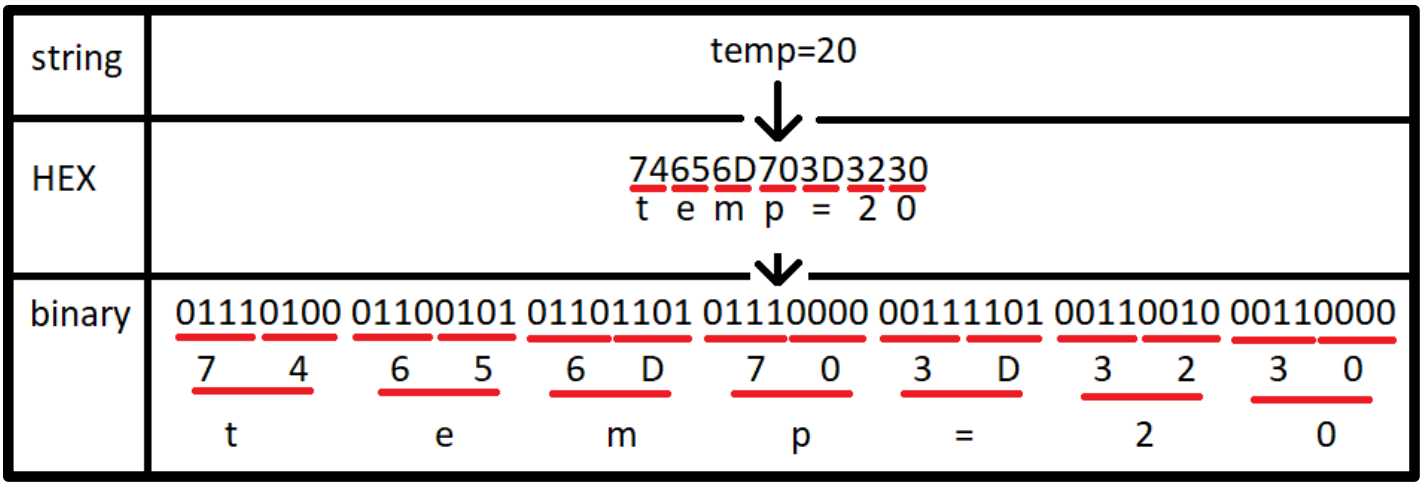

Рисунок 1. Пример кодирования сообщения “temp=20”

Этап № 2. 3D-моделирование прототипа МКА формата Cubesat 1U

С учётом требований команде необходимо разработать трёхмерную модель прототипа МКА формата Cubesat 1U, включающую в себя трёхмерные модели модулей, элементов МКА и корпуса.

В рамках выполнения этапа № 2 команде участников необходимо:

1. Разработать 3D-модель корпуса МКА в соответствии с требованиями.

Требования к 3D-модели корпуса:

- Корпус представляет из себя куб размерами 10 × 10 × 10 см, без выходящих за его пределы элементов конструкции или содержимого.
- Корпус должен обладать креплениями под все перечисленные внутренние элементы.
- Крепления должны иметь возможность соединения с элементами посредством болта и гайки.
- Каждый элемент должен крепиться к корпусу или к элементу, крепящемуся к корпусу, не менее чем в двух точках. Соединительные провода должны пролегать вдоль конструкции и крепиться к ней.
- В конструкции должна присутствовать возможность замены любого из внутренних модулей (элементы можно достать, не нарушая целостности элементов корпуса МКА).

2. Разработать 3D-модели модулей и элементов МКА в соответствии с требованиями.

Требования к 3D-модели модулей и элементов МКА:

- 4 солнечные панели размером не более 90 × 90 × 2мм, креплениями для соединения с корпусом МКА, возможностью соединения с другими элементами МКА при помощи соединительных проводов.
- Блок аккумуляторной батареи размеров не более 90 × 80 × 25мм и креплениями для соединения с корпусом МКА и возможностью соединения с другими элементами МКА при помощи соединительных проводов.
- Датчик температуры LM35 и датчик Холла KY-024 должны быть интегрированы в корпус МКА и измерять температуру окружающей среды вне радиомодуля. 3D-модель датчика температуры LM35 необходимо реализовать в соответствии с документацией, находящейся в Приложении № 3.
- Соединительные провода: диаметр провода 1 мм.

Этап №3. Дублирование устройства и функционала” чёрного ящика”

На данном этапе требуется разработать устройство, оснащённое датчиком магнитного холла и датчиком температуры, которое формирует сообщения, соответствующие радиосообщениям, отправляемым с “чёрного ящика”, выводит их на LCD-дисплей.

Перечень необходимых элементов: микроконтроллер Arduino UNO, датчик холла KY-024, датчик температуры LM35, LCD-дисплей, провода, резисторы (Приложение № 4).

Участникам требуется:

- Собрать схему подключения устройства.
- Разработать программный код для микроконтроллера Arduino UNO, реализующий снятие показаний с датчиков и вывод их на экран через serial port.
- Реализовать расчёт температуры и интенсивности магнитного поля, на основе показаний датчика в соответствии с приложением № 4.
- Реализовать отображение полученных показаний с датчика в виде раскодированных сообщений, полученных с “чёрного ящика” на этапе 1 (реализовывать кодирования сообщения и отправку по радио не требуется).
- Реализовать вывод сообщения на LCD-дисплей.

Приложение 1. Сообщение с чёрного ящика

6d6573736167655f73746172745f5f
4853455f5341545f323032315f5f5f
74656d705f646174613d36395f5f5f
6669656c645f646174613d3736385f
6d6573736167655f656e645f5f5f5f

Приложение 2. Кодирование ASCII

ASCII – способ представления символов (в том числе букв) двоичными числами. Каждому символу, в том числе каждой букве, ставится в соответствие восьмиразрядное двоичное число, то есть байт. Поскольку байт можно представить в виде двух шестнадцатеричных цифр (старшие четыре бита и младшие четыре бита), коды символов нередко записывают в шестнадцатеричном представлении – для экономии места. Таблица перевода из кодировки ASCII в шестнадцатеричную систему представлена на рисунке 1.

Далее пример перевода сообщения из двоичного представления в текст, с использованием шестнадцатеричного представления, как промежуточного. Пусть дано сообщение: “temp=20”. Разобьём его на буквы: ‘t’, ‘e’, ‘m’, ‘p’, ‘=’, ‘2’, ‘0’. По таблице на рисунке 1 переведём буквы в шестнадцатеричное представление, и после – в двоичное. Результаты представлены в таблице 1.

Таблица 1.

Символ	**t**	**e**	**m**	**p**	**=**	**2**	**0**
Шестнадцатерич-ное представление	74	65	6D	70	3D	32	30
Двоичное представление	01110100	01100101	01101011	01110000	00111011	00110010	00110000

Итого получаем: “01110100 01100101 01101011 01110000 00111011 00110010 00110000”. Процесс преобразования двоичного представления в текст происходит в обратном порядке.

ASCII	HEX	ASCII	HEX	ASCII	HEX	ASCII	HEX
NUL	00	DEL	10	Space	20	0	30
SOH	01	DC1	11	!	21	1	31
STX	02	DC2	12	″	22	2	32
ETX	03	DC3	13	#	23	3	33
EOT	04	DC4	14	$	24	4	34
ENQ	05	NAK	15	%	25	5	35
ACK	06	SYN	16	&	26	6	36
BEL	07	ETB	17		27	7	37
BS	08	CAN	18	(	28	8	38
SH	09	EM	19	)	29	9	39
LF	0A	SUB	1A	*	2A	:	3A
VT	0B	ESC	1B	+	2B	;	3B
FF	0C	FS	1C	,	2C	<	3C
CR	0D	GS	1D	-	2D	=	3D
SO	0E	RS	1E	.	2E	>	3E
SI	0F	US	1F	/	2F	?	3F

ASCII	HEX	ASCII	HEX	ASCII	HEX	ASCII	HEX
@	40	P	50		60	p	70
A	41	Q	51	a	61	q	71
B	42	R	52	b	62	r	72
C	43	S	53	c	63	s	73
D	44	T	54	d	64	t	74
E	45	U	55	e	65	u	75
F	46	V	56	f	66	v	76
G	47	W	57	g	67	w	77
H	48	X	58	h	68	x	78
I	49	Y	59	i	69	y	79
J	4A	Z	5A	j	6A	z	7A
K	4B	[	5B	k	6B	{	7B
L	4C	\	5C	l	6C	|	7C
M	4D	]	5D	m	6D	}	7D
N	4E	^	5E	n	6E	~	7E
O	4F	_	5F	o	6F	DELL	7F

Рисунок 1. Таблица кодирования ASCII

Приложение 4. Краткая техническая спецификация датчиков.

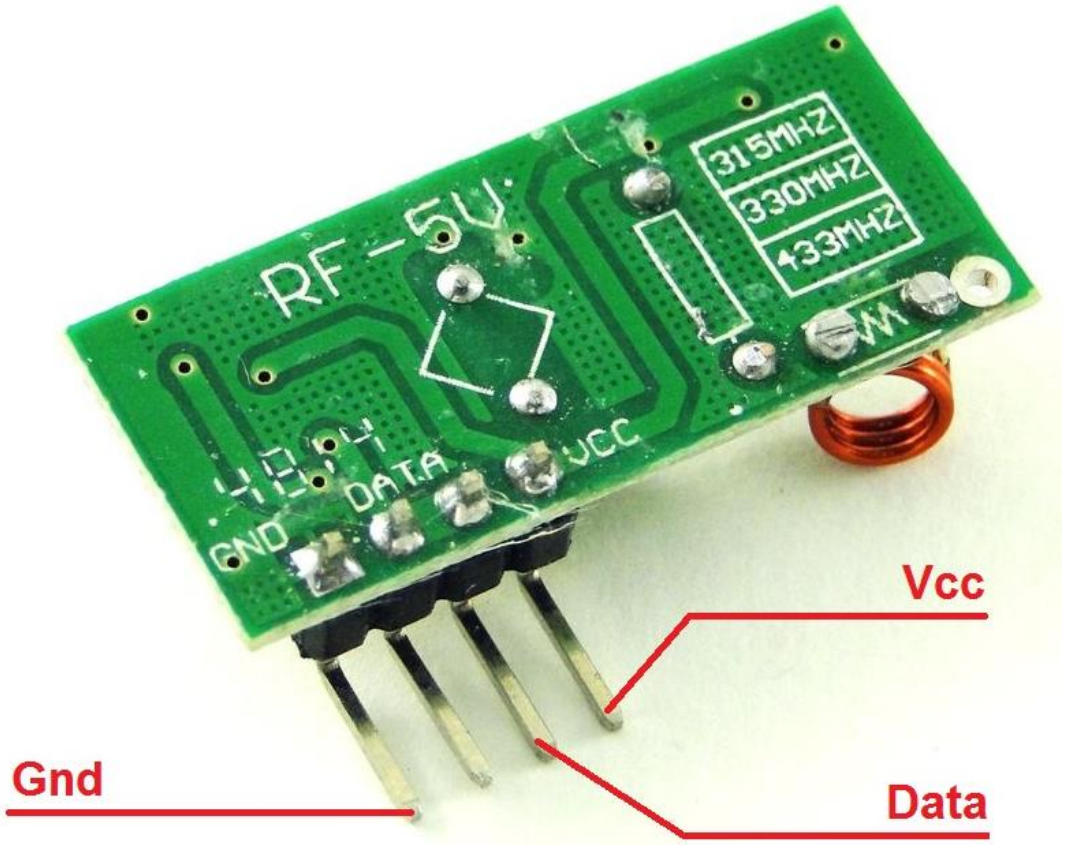

Рисунок 1. Радиоприёмник XD-RF-5V 433МГц, наименование выводов.

Технические характеристики:

- Рабочая частота: 433 МГц.
- Напряжение питания 5V.

Для написания программного кода требуется использовать библиотеки RH_ASK.h и SPI.h.

Для создания объекта, ответственного за приёмник, требуется использовать команду RH_ASK driver;

Команда driver.init() – производит инициализацию приёмника и возвращает правду при успехе.

Команда driver.recv(buf, &buflen) – производит считывание сообщения и запись его по указателю в массив buf размером buflen (buf и buflen – целочисленное 8-битное unsigned число).

Для верной работы приёмника его требуется подключить к 11 цифровому выходу платы.

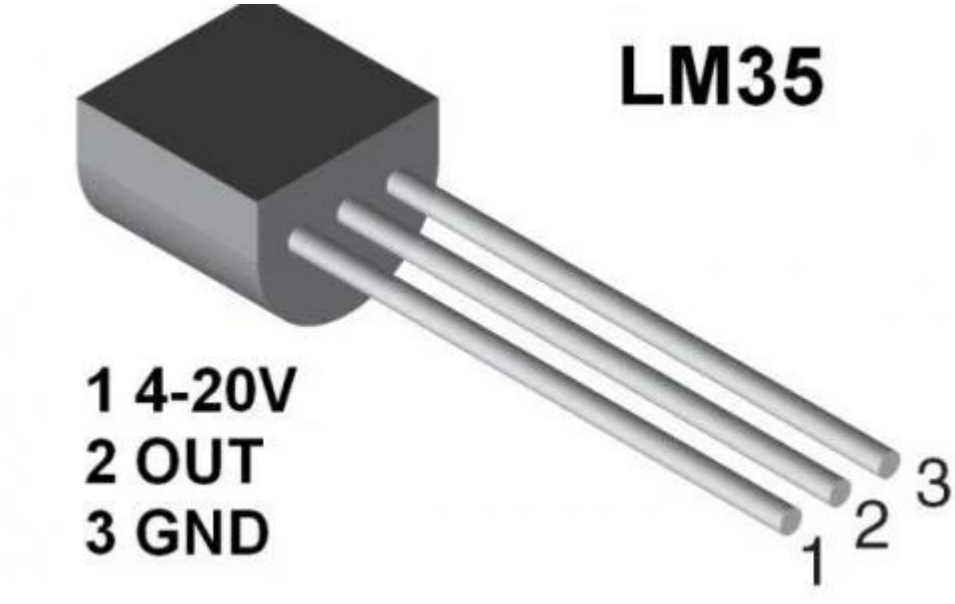

Рисунок 2. Датчик температуры lm35, наименование выводов.

Технические характеристики:

- Диапазон температур: 0 … 150 °C.
- Напряжение питания: от 4.0 до 20 В.

На выходе датчика формируется напряжение пропорционально температуре по шкале Цельсия.

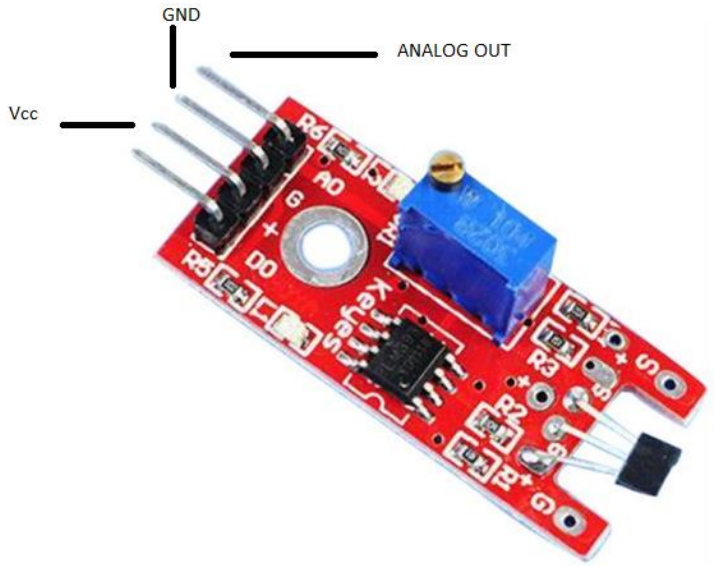

Рисунок 3. Датчик холла KY-021, наименование выводов.

Технические характеристики:

- Магнитный диапазон: от ±650 до ±1000 Гс.
- Чувствительность: 1,4 мВ/Гс.
- Напряжение питания: 5 В.

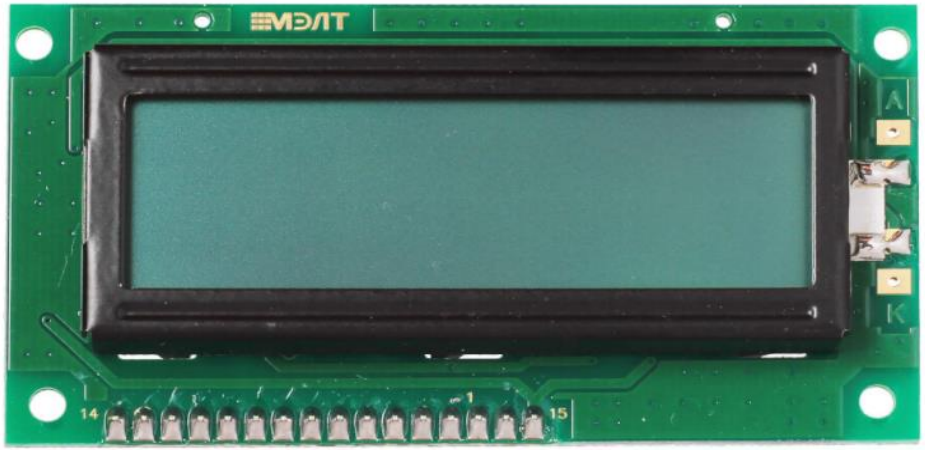

Рисунок 4. LCD-дисплей МЭЛТ 16x2.

Таблица 1. Наименование выводов LCD-дисплея

№	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Обозначение	GND	Vcc	VO	RS	R/W	E	DB0	DB1	DB2	DB3	DB4	DB5	DB7	DB7	Vcc	GND

Напряжение питания: 5 В.
Для работы с LCD-дисплеем на Arduino используется встроенная библиотека LiquidCrystal.h.

В данной библиотеке для инициализации объекта-дисплея используется команда:
LiquidCrystal lcd(RS, E, DB4, DB5, DB6, DB7).

Для работы с дисплеем устанавливается его размер командой:
lcd.begin(x, y); где *y* – число строк, а *x* – число символов на одной строке;
lcd.print("Hello world"); используется для вывода на дисплей;
lcd.setCursor(x, y); используется для перестановки курсора, *y* – номер строки, а *x* – номер символа;
lcd.clear(); используется для очистки дисплея.

Приложение 3. Чертежи для моделирования

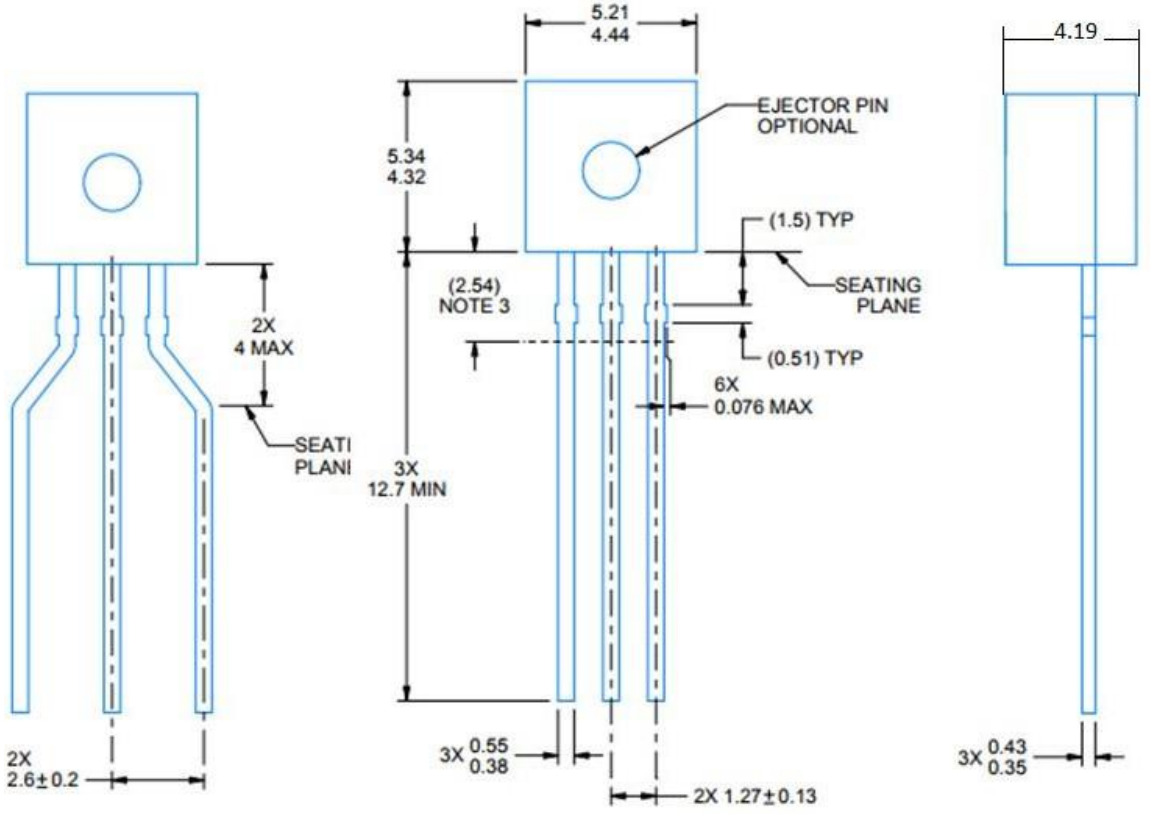

Рисунок 1. 2D - модель датчика LN35 (размеры в миллиметрах)

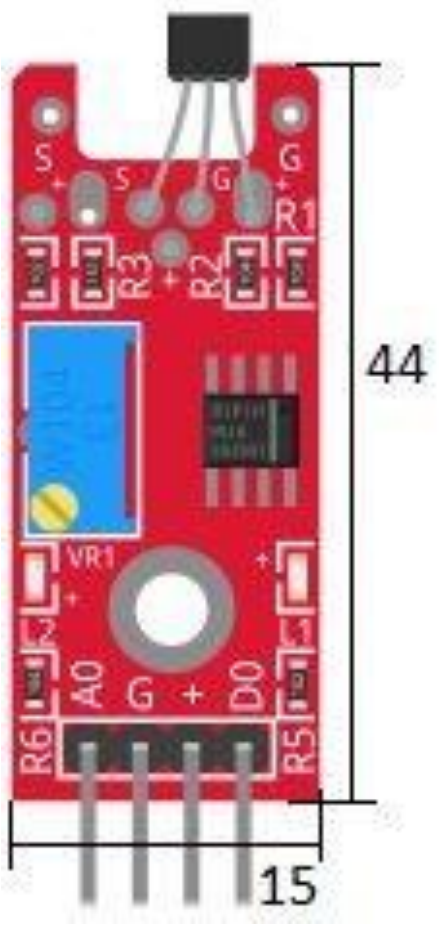

Рисунок 2. 2D - модель датчика KY-024 (размеры в миллиметрах).

Высота датчика13 мм