

১২/ আল্ট্রাসনিক সেন্সর দিয়ে দূরত্ব পরিমাপ করে ডিসপ্লে তে দেখাও :

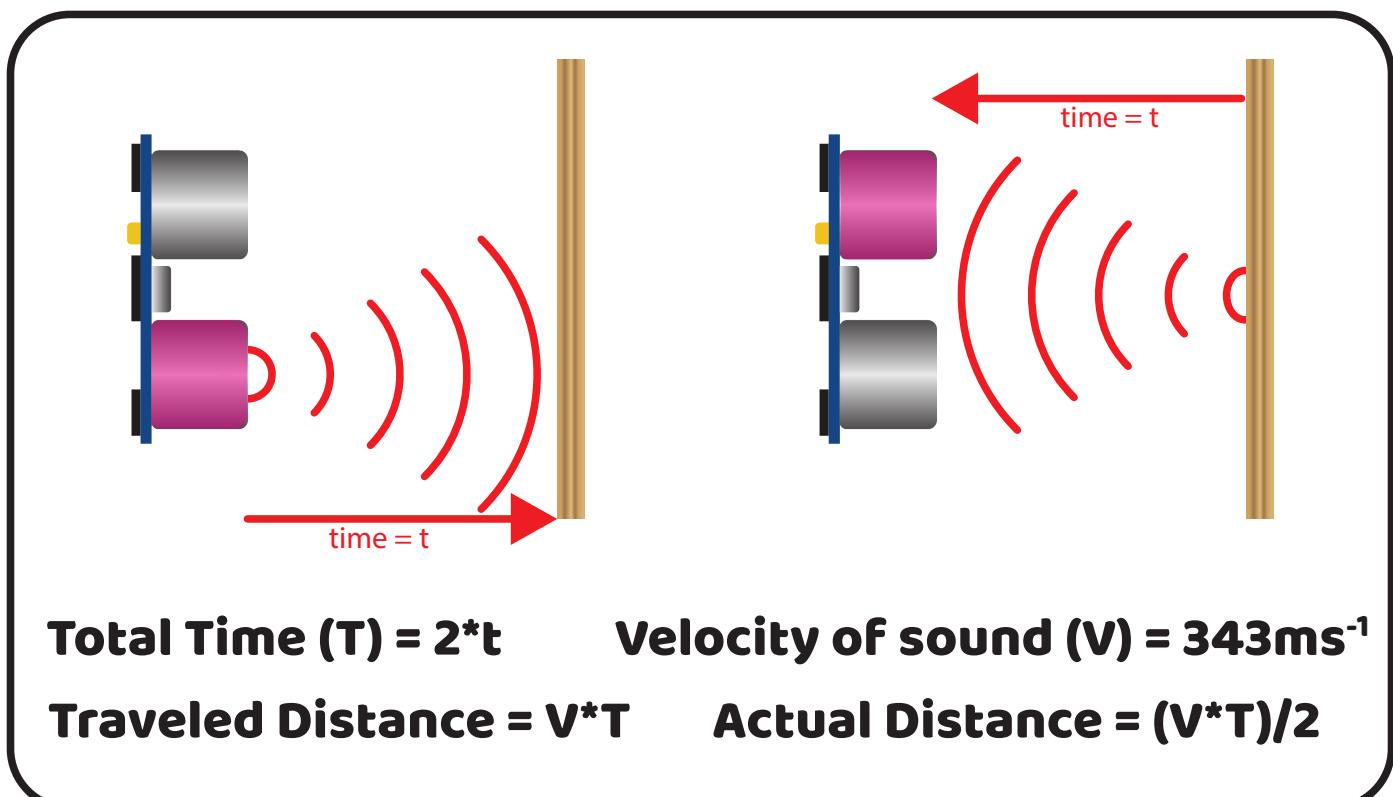
প্রয়োজনীয় উপকরণ:

- ১/ ডিসপ্লে মডিউল
- ২/ ব্রেডবোর্ড
- ৩/ জাম্পার ওয়্যার
- ৪/ JRC বোর্ড
- ৫/ আল্ট্রাসনিক সেন্সর
- ৬/ ২২০ ওহম রেজিস্টর

বর্ণনা: এই প্রজেক্টে আমরা একটি আল্ট্রাসনিক সেন্সর দিয়ে কোন বস্তুর দূরত্ব পরিমাপ করে এরপর সেই মান আগের ন্যায় ডিসপ্লে মডিউলে দেখানোর ব্যবস্থা করবো। এখানে আমাদের প্রথমেই জানা প্রয়োজন আল্ট্রাসনিক সেন্সর আসলে কি।

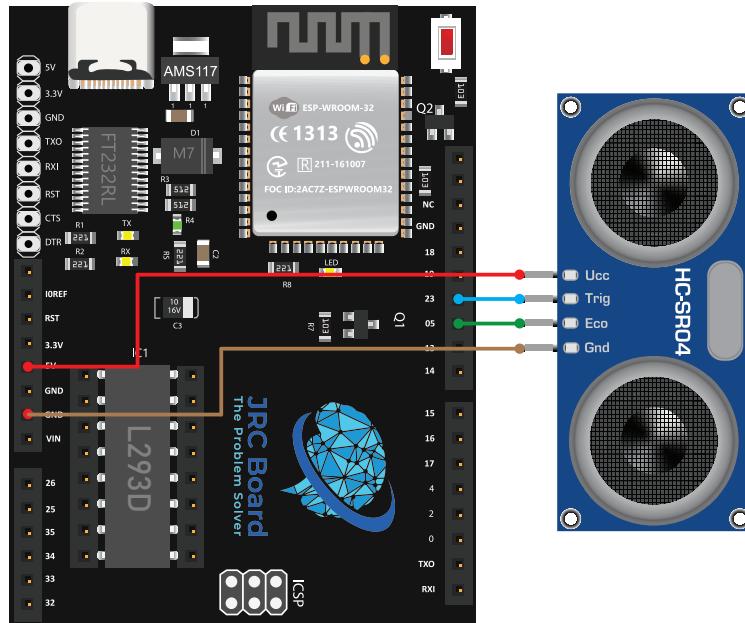


আল্ট্রাসনিক সেন্সর আসলে একটি মডিউল যেটা অতি উচ্চ কম্পাক্ষের শব্দ পাঠায় এবং সেটা কোন বস্তুর গায়ে প্রতিফলিত হয়ে ফিরে আসলে মধ্যবর্তী সময় পরিমাপ করে তা থেকে দূরত্ব নির্গত করে দেখায়। এখানে দুটি মাথা কিংবা চোখ দেখা যায়। একটা হলো ট্রান্সমিটার এবং অপরটি রিসিভার। ট্রান্সমিটার দিয়ে উচ্চ কম্পাক্ষের শব্দ নির্গত হয় যা আমরা কানে শুনতে পারিনা। এরপর সেটা কোন বস্তুতে বাধাপ্রাপ্ত হয়ে ফিরে আসলে সেটা রিসিভার প্রান্তের পর্দা আন্দোলিত করে যেটা দেখে সেন্সর বুঝতে পারে যে কত সময় পরে শব্দ ফিরে এসেছে।



ছবিতে দেখতে পাচ্ছো যে কিভাবে সে শব্দ পাঠায় এবং এখানে এটি শব্দ ফিরে আসার পরে শব্দ পাঠানোর এবং ফিরে আসার মধ্যবর্তী সময় পরিমাপ করে সেটা JRC বোর্ডে পাঠাবে এবং পরবর্তীতে আমরা সেই সময়কে উপরোক্ত হিসেবের মাধ্যমে দূরত্ব পরিমাপ করে এরপর সেটা আউটপুট হিসেবে প্রদর্শন করি।

এখানে লক্ষ্য করলে দেখতে পাবে যে এখানে সেপরে মোট 8টি পিন রয়েছে। এর মধ্যে দুটি হলো পাওয়ার পিন (VCC ও GND) যেগুলো আমরা JRC বোর্ডের 5V ও GND পিনে সংযোগ দিবে। বাকি দুটি পিন হলো TRIG পিন এবং ECO পিন। এখানে TRIG পিনে নির্দিষ্ট সিগ্নাল দিয়ে অল্ট্রাসনিক সেন্সর এর ট্রান্সমিটার কে শব্দ পাঠাতে বলা হয়। এবং সেই শব্দ ফিরে আসলে সেটি কর সময় পরে ফিরে এসেছে সেটা ECO পিন জানিয়ে দেয়। এই দুইটি পিন JRC বোর্ডের যেকোন ডিজিটাল পিনের সাথে সংযোগ দিলেই হয়ে যায়। নিচের চিত্রের দিকে লক্ষ্য করিঃ



এখানে TRIG পিনের সাথে JRC বোর্ডের 23 নাম্বার পিন এবং ECO পিনের সাথে JRC বোর্ডের 05 নাম্বার পিনের সাথে সংযোগ করা হয়েছে। এবার আমাদের কোডিং করার পালা। নিচের চিত্রের দিকে লক্ষ্য করোঃ

```

int trig = 23, echo = 05;
void setup() {
    pinMode(trig, OUTPUT);
    pinMode(echo, INPUT);
    Serial.begin(9600);
}
void loop() {
    int x = reading();
    Serial.println(x);
    delay(50);
}
int reading() {
    digitalWrite(trig, LOW);
    delayMicroseconds(2);
    digitalWrite(trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(trig, LOW);
    long duration = pulseIn(echo, HIGH);
    int distance= duration*0.0343/2;
    return distance;
}

```

কোডের ব্যাখ্যায় আসি। শুরুতেই এখানে TRIG পিন এর ভূমিকা নির্ধারণ করে দিতে হচ্ছে। যেহেতু আমরা TRIG পিন দিয়ে সিগ্নাল পাঠিয়ে শব্দ পাঠানো শুরু করতে চাই, সেক্ষেত্রে TRIG পিনের ভূমিকা হবে আউটপুট এবং ECO পিন দিয়ে রিডিং নিছিঁ বলে এর ভূমিকা হলো ইনপুট যা setuo() ফাংশনের ভেতরে কিখে দেয়া হয়েছে। সিরিয়াল মনিটরে দূরত্ব দেখাবো বলে আগেই Serial.begin(9600) লিখে নেয়া হয়েছে। loop() ফাংশনে যাবার আগে আমরা নিচে ম্যানুয়াল ফাংশন "reading()" এর দিকে তাকাই। ম্যানুয়াল ফাংশনের কাজ হলো এর মধ্যে কিছু স্টেটমেন্ট লিপিবদ্ধ করে রাখা যা কিনা এই ফাংশন কল করামাত্র সম্পাদন করে নিবে। একটা ছেট উদাহরণ দিই। ধরো তুমি তোমার দারোয়ান কে বলে রেখেছো যে গাড়ির হর্ণ দিলে যাতে গেট খুলে দেয়। এখানে এই যে ইন্সট্রাকশন দিয়ে রাখা হয়েছে সেটা হলো statement। এখন তুমি যদি এসে গাড়ির হর্ণ বাজাও, দারোয়ান গেট খুলে দিবে, মানে তাকে যে কাজ বুঝিয়ে দেয়া হয়েছিলো, সে ঐ কাজ সম্পাদন করা করে ফেলেছে। এই যে গাড়ির হর্ণ দিয়ে ইঙ্গিত দিলে, এটাই হচ্ছে ম্যানুয়াল ফাংশন কল করা। দেখতে পাচ্ছো যে loop() ফাংশনের ভেতরে reading(); লেখা আছে, এই লেখার মাধ্যমেই ম্যানুয়াল ফাংশন কল করা হয় এবং একবার কল করলে সেই ফাংশনের ভেতরে যে স্টেটমেন্ট দেয়া আছে সেটা সে সম্পাদন করে ফেলে। এখন চল দেখা যাক এর ভেতরে আসলে কি কি ইন্সট্রাকশন অর্থাৎ স্টেটমেন্ট দেয়া আছে।

এই ফাংশনের শুরুতে `int reading()` লেখা কারণ আমরা এই ফাংশনের কাজ শেষে এখান থেকে কোন একটা ভ্যালু আউটপুট নিবো (দূরত্বের মান)। এখন এই ফাংশনের ভেতরে শুরুতে `trig` পিনের স্টেট লো করে নেয়া হয়েছে এবং অন্তত ২ মাইক্রোসেকেন্ড এই অবস্থায় রাখা হয়েছে। এভাবে আসলে এই সেন্সর কে প্রস্তুত করা হয় আসল কাজ শুরুর জন্য, যেমনটা তুমি দেখে থাকো দৌড় শুরুর আগে খেলোয়াড় যেভাবে প্রস্তুত নেয়। এরপর `trig` পিনে ১০ মাইক্রোসেকেন্ড যাবৎ হাই স্টেটে রেখে এরপর আবার তার স্টেট লো করে দেয়া হয়। এর ফলে `trig` পিনে ১০ মাইক্রোসেকেন্ড যাবৎ একটা পালস পায় যেটা কিনা সেন্সরের কাজ শুরুর ইঙ্গিত প্রদান করে।

এরপরে সেন্সর আল্ট্রাসনিক সাউন্ড পাঠ্য সামনের দিকে এবং ফেরত আসা পর্যন্ত অপেক্ষা করে। যখনই ফেরত আসে, তখনই `eco` পিনে ৫ ভোল্ট আসতে শুরু করে। এবং মজার কথা হলো এই শব্দ যত সময় পরে ফিরে এসেছে, ঠিক তত সময় ধরে এই পিনে ৫ ভোল্ট পেতে থাকবে। যেমন যদি শব্দটি এক সেকেন্ড পরে ফিরে আসে, সেক্ষেত্রে এই `eco` পিন এক সেকেন্ড যাবৎ অন হয়ে থাকবে, এরপর সে অফ হয়ে যাবে।

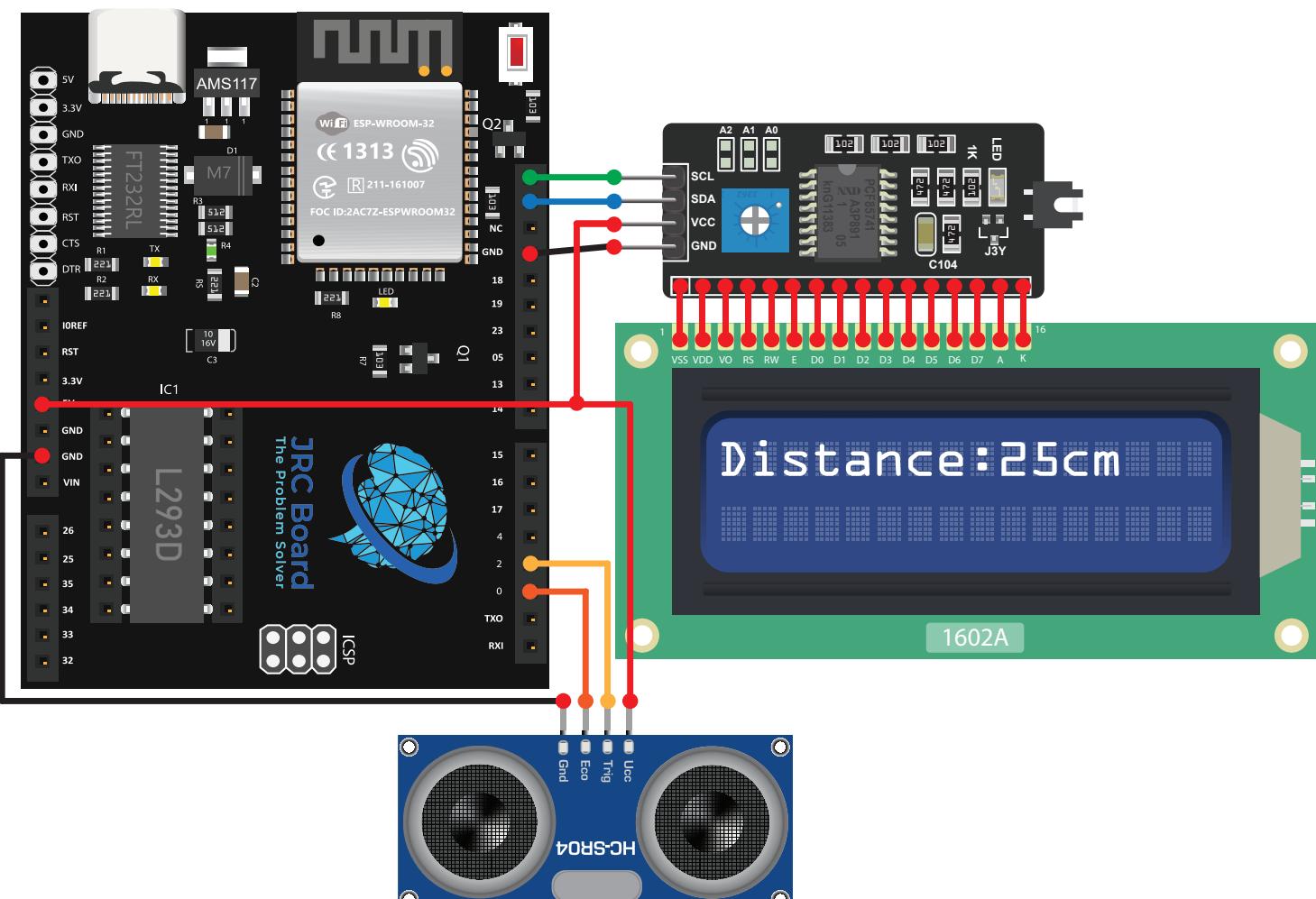
তার মানে আমাদের এটা মেপে দেখতে হবে যে এই পিন কত সময় ধরে অন ছিলো তাইনা? এটির জন্য আমরা `pulseIn()` ফাংশন ব্যবহার করে থাকি। এর ভেতরে শুরুতে কোন পিন থেকে রিডিং নিতে চাই সেটা লিখে দিই (এই প্রজেক্টের বেলায় যেটা কিনা `echo` ভ্যারিয়েবল এর মান)। এবং কমা দিয়ে এরপর আমরা এটা লিখি যে কোন অবস্থার সময় পরিমাপ করতে চাই। যেমন আমাদের এই প্রজেক্টে আমরা `echo` পিন টি কত সময় ধরে অন ছিলো সেটা মাপতে চাই, তাই এখানে আমরা `pulseIn(echo, HIGH)`; লিখে দিচ্ছি।

উল্লেখ্য যে `pulseIn()` ফাংশন থেকে যে আউটপুট আসে সেটা হলো মাইক্রোসেকেন্ডে যার মান অনেক বড়ো হতে পারে। যে কারণে আমরা যে ভ্যারিয়েবলে এর মান ধারণ করছি (প্রজেক্টের বেলায় `duration`) সেখানে `int` টাইপের বদলে `long` টাইপ ভ্যারিয়েবল ব্যবহার করেছি যার ফলে ৪বাইট ধারণক্ষমতা থাকে এর ভেতরে।

যেহেতু এখানে আমরা মাইক্রোসেকেন্ড এককে মান পাচ্ছি, সেটাকে সূত্রে হিসেব করার জন্য অবশ্যই সেকেন্ডে রূপান্তর করতে হবে। সেজন্যে যে মান পাচ্ছি সেটিকে আমরা অবশ্যই ১০০০০০০ দিয়ে ভাগ করে নিবো। বাতাসে শব্দের গতিবেগ সাধারণ অবস্থায় আমরা জানি প্রতি সেকেন্ডে ৩৪৩ মিটার। যদি আমরা সেন্টিমিটার এককে দূরত্ব পেতে চাই (মিটার একক খুবই বড় দূরত্ব পরিমাপের জন্য), সেক্ষেত্রে এখানে বলতে পারি বাতাসে শব্দের গতিবেগ প্রতি সেকেন্ডে ৩৪৩০০ সেমি। যদি এটা মাইক্রোসেকেন্ডে হিসেব করতে যাই, সেক্ষেত্রে দাঁড়ায় 0.0343 মিটার প্রতি মাইক্রোসেকেন্ড (মাইক্রোসেকেন্ডে মাপছি কারণ আমরা `pulseIn()` ফাংশন থেকে মাইক্রোসেকেন্ড এককে আউটপুট পাচ্ছি)।

বেগের এই মানটিই ব্যবহার করা হয়েছে পরবর্তী লাইনের সমীকরণে। এখানে দূরত্ব $= ((\text{সময়} * \text{বেগ}) / 2)$ ব্যবহার করা হয়েছে কারণ শব্দ তো গিয়ে আবার ফিরে আসছে, এর মানে এখানে শব্দ আমাদের কাঞ্জিত দূরত্ব দুইবার অতিক্রম করছে যে কারণে আসল দূরত্ব পেতে ২ দিয়ে ভাগ করা হয়েছে। এভাবেই আমরা আল্ট্রাসনিক সেন্সর এর রিডিং থেকে নির্ণয় দূরত্ব বের করে নিতে পারি। যে মান টি পাচ্ছি সেটি আমরা `distance` নামক ভ্যারিয়েবলে সেভ করে নিচ্ছি এবং এই ভ্যারিয়েবলের মানটিই `return distance;` এর মাধ্যমে ফাংশন থেকে রিটার্ন করছি। অর্থাৎ আমরা এই ফাংশন যখনই কল করবো, এই ফাংশন নিজের কাজ শেষে দূরত্বের মান আউটপুট দিয়ে দিবে।

সুতরাং, আমরা `loop()` ফাংশনের ভেতরে `int x = reading();`; লেখার মাধ্যমে প্রথমে `reading` ফাংশনটিকে কল করছি, এবং এখান থেকে যে আউটপুট আসে সেটিকে `x` ভ্যারিয়েবলে সংরক্ষণ করে নিচ্ছি। এবং এই মানটিই সিরিয়াল মনিটরে দেখিয়ে দিচ্ছি। এর ফলে আমাদের আল্ট্রাসনিক সেন্সর দিয়ে দূরত্ব পরিমাপ করে সেটি কম্পিউটার স্ক্রিনে প্রদর্শন করার ব্যবস্থা হয়ে গেছে। এবার কাজ হচ্ছে ডিসপ্লে মডিউলে এই দূরত্বের মান প্রদর্শন করার ব্যবস্থা করা। আমরা ইতিমধ্যেই জানি যে ডিসপ্লে মডিউলে I²C কনভার্টারের মাধ্যমে JRC বোর্ডের সাথে সংযোগ দেয়া যায়। সেখানে একটা আলাদা করে সোনার সেন্সর বসালেই কিন্তু কাজ হয়ে যাবে। নিচে এটি দেখানো হচ্ছে:



দুই কম্পোনেন্ট একসাথে করে ফেলা মানে দুইটার কোডও বুদ্ধি খাটিয়ে একসাথে করে ফেলা তাইনা? সেক্ষেত্রে কোডের কাঠামো নিচের মতো দাঁড়ায়:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
int trig = 2, echo = 0;

void setup() {
    pinMode(trig, OUTPUT);
    pinMode(echo, INPUT);
    Serial.begin(9600);
    lcd.init(); lcd.backlight();
}

void loop() {
    lcd.setCursor(0,1);
    lcd.print("Distance:");
    int x = reading();
    lcd.print(String(x));
    lcd.print("cm   ");
    delay(50);
}

int reading() {
    digitalWrite(trig, LOW);
    delayMicroseconds(2);
    digitalWrite(trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(trig, LOW);
    long duration = pulseIn(echo, HIGH);
    int distance = duration * 0.0343 / 2;
    return distance;
}
```

এখানে কিছু ব্যাপার লক্ষ্যনীয় যে আমরা যেহেতু এখন ভিন্ন পিনে সোনার সেন্সর এর trig এবং eco পিনের সংযোগ দিয়েছি, তাই কোডে trig এবং eco ভ্যারিয়েবলের মান পরিবর্তন করে নেয়া হয়েছে। এবং আরেকটি ব্যাপার হলো lcd ডিসপ্লে তে লেখা দেখানোর জন্য lcd.print() ফাংশন যোটি ব্যবহার করছি সেটিতে কোন ভ্যারিয়েবলের মান সরাসরি দেখানো যায়না। কারণ ভ্যারিয়েবল এর মান আসলে একটি সংখ্যা, এটি কোন লেখা নয়। এটিকে লেখা হিসেবে রূপান্তর করার জন্য String() ফাংশনটি ব্যবহার করে থাকি। এই ফাংশনের ভেতরে যেকোন নাম্বার দিলে সেটাকে সে লেখায় রূপান্তর করে ফেলে এবং সেই লেখা টি পরবর্তীতে ডিসপ্লে তে দেখানোর ব্যবস্থা করা হয়েছে। এর ফলে আমরা কিন্তু আল্ট্রাসনিক সেন্সর থেকে পাওয়া দূরত্বের মান আরামসে ডিসপ্লে মডিউলে দেখাতে পারছি।

উল্লেখ্য যে আল্ট্রাসনিক সেন্সর থেকে দূরত্ব নির্ণয়ের প্রসেস টি একটু সময় নেয় এবং মাঝে অল্প একটু ডিলে না দিলে এটা পারফেক্টলি কাজ করতে পারেনা। এবং এটার দূরত্ব নির্ণয়ের একটি সীমা রয়েছে যার মান হচ্ছে 3 মিটার। এবং এটা দিয়ে যে বক্ষত দূরত্ব মাপা হবে তার তল মসৃণ হওয়া এবং সেন্সর এর মুখোমুখি হওয়া জরুরি।