

১১/ ডিসপ্লে মডিউল এ লেখা দেখাও :

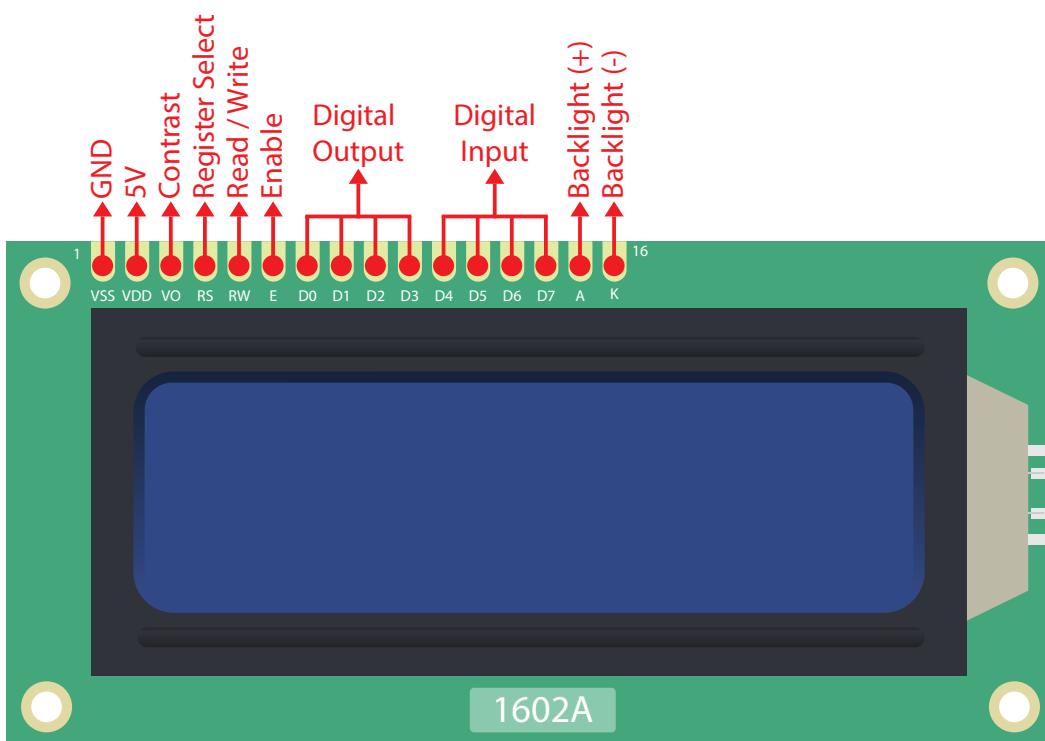
প্রয়োজনীয় উপকরণ:

- ১/ ডিসপ্লে মডিউল
- ২/ ব্রেডবোর্ড
- ৩/ জাম্পার ওয়্যার
- ৪/ JRC বোর্ড
- ৫/ I2C কনভার্টার
- ৬/ ২২০ ওহম রেজিস্ট্র

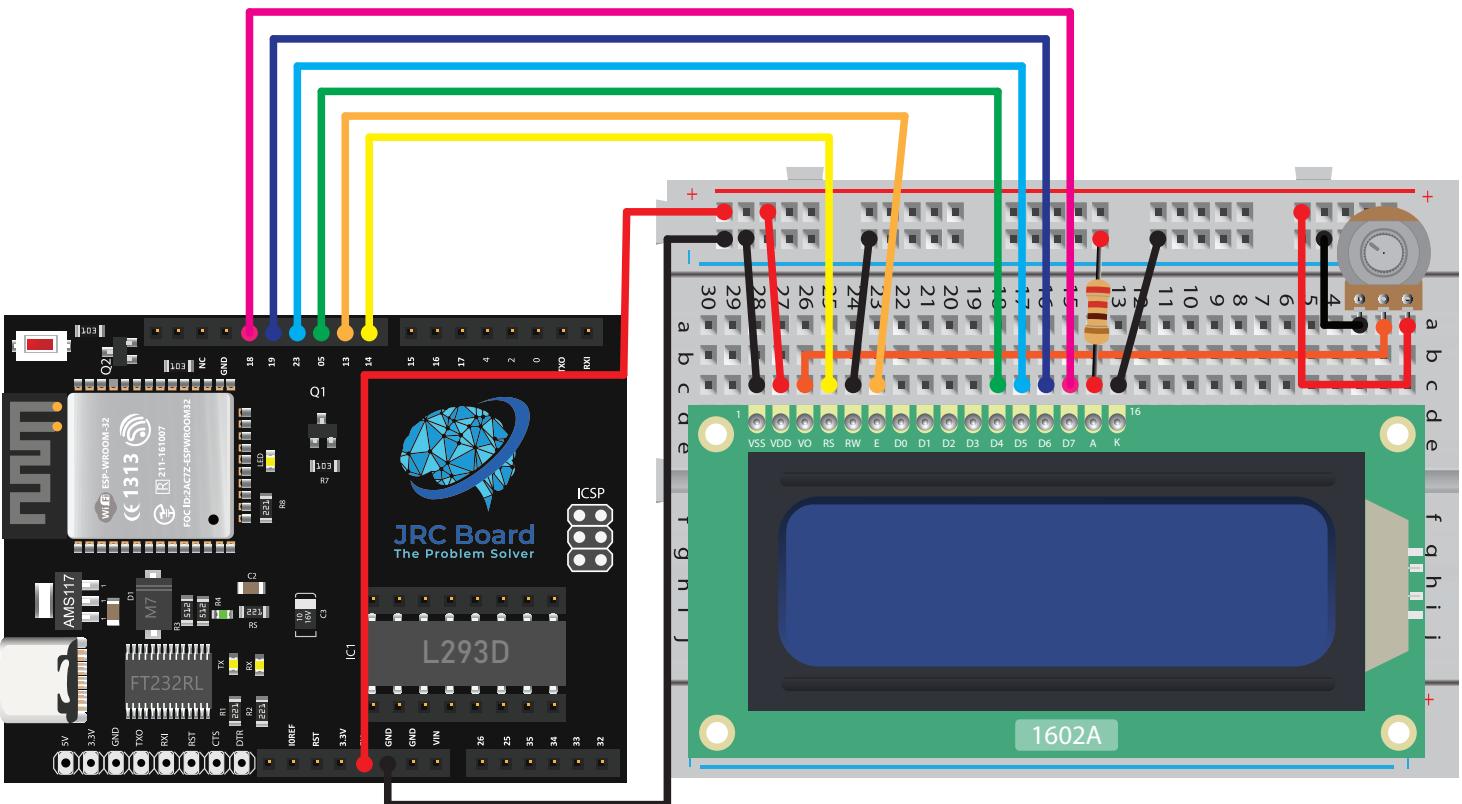
বর্ণনা: এই প্রজেক্টে আমরা একটি LCD ডিসপ্লে তে JRC বোর্ড থেকে লেখা দেখানো যায় সেটি শিখবো। প্রথমেই জেনে নেয়া যাক এই ডিসপ্লে মডিউল আসলে কি।



এটিই হচ্ছে সেই এলসিডি ডিসপ্লে মডিউল যেখানে আমরা যথাযথ সিগ্নাল দিলে এটি স্ক্রিনে লেখা প্রদর্শন করবে। বাজারে অনেক ধরণেরই ডিসপ্লে মডিউল পাওয়া যায় যার মধ্যে এটি সবথেকে সস্তা এবং ব্যবহার করা তুলনামূলক সহজ। এখানে দুই লাইনে লেখা প্রদর্শন করা যায় এবং প্রতি লাইনে 16 টি অক্ষর জায়গা হয়। এর চেয়েও বড় সাইজে পাওয়া যায়, কিন্তু আমরা আপাতত এটিতেই কাজ চালিয়ে নিতে পারি। এখানে মোট 16 টি পিন রয়েছে। পিন এর মার্কিং নিচে দেখানো হচ্ছে:



এখানে একদম শুরুতে রয়েছে VSS ও VDD পিন যেগুলো JRC বোর্ডে যথাক্রমে GND এবং 5V পিনে লাগাতে হয় এন্টিভেট করার জন্য। এর পরেই রয়েছে ডিসপ্লে তে যে লেখা দেখাবে তার কন্ট্রাস্ট নিয়ন্ত্রণ এর জন্য একটি পিন যেখানে 0 থেকে 5 ভোল্টের মধ্যে যেকোন ভোল্টেজ ইনপুট দিলে এর কন্ট্রাস্ট বাড়ে বা কমে। এরপর রয়েছে RS পিন বা রেজিস্টার সিলেক্ট পিন। এতে ডিজিটাল সিগ্নাল দিতে হয়। 0 দিলে কমান্ড মুডে থাকে এবং 1 দিলে ডেটা মুডে থাকে। কমান্ড মুডের কাজ হলো স্ক্রিন ক্লিয়ার করা, কার্সর সরানো ইত্যাদি। আর যদি কোন অক্ষর দেখাতে চাও, তবে এই পিনে অবশ্যই ডিজিটাল 1 অর্থাৎ 5 ভোল্ট দিয়ে ডেটা মুডে নিতে হয়। এই ডিসপ্লে তে চাইলে আমরা কোন কমান্ড দিতে পারি অথবা চাইলে এই ডিসপ্লে থেকে আমরা ফিডব্যাক নিতেও পারি। এই ব্যাপার টি নিয়ন্ত্রন করা যায় RW পিন বা Read/Write পিন দিয়ে। এই পিনে ডিজিটাল 0 দিয়ে কমান্ড ইনপুট দেয়া যায়, অথবা চাইলে ডিজিটাল 1 প্রদান করে মডিউল থেকে ফিডব্যাক পাওয়া যায়। যেহেতু আমরা কেবল এতে অক্ষর ইনপুট দিয়েই যাবো, তাই আমরা এই পিনের সাথে সবসময় GND পিনের সংযোগ দিয়ে রাখতে পারি। এর পরে রয়েছে Enable পিন যেটিতে ডিজিটাল 1 প্রদান করলে আমাদের যাবতীয় কমান্ড ডিসপ্লে মডিউল কাজ করে দেখায়। এরপর রয়েছে 8 টি ডেটা লাইন রয়েছে যেখানে প্রথম 4টি থেকে ফিডব্যাক পাই এবং পরের চারটিতে ডিজিটাল সিগ্নাল প্রদান করে প্রযোজনীয় অক্ষর ইনপুট দেই। যেহেতু আমাদের ফিডব্যাক নিয়ে মাথা না ঘামালেও চলে, তাই আমরা কেবল পরের 4 টি ব্যবহার করবো। শেষের দুটি পিন হলো এলসিডি মডিউলের ব্যাকলাইট জ্বালানোর জন্য। এখানে A পিনে পেজেটিভ 5V এবং K পিনে নেগেটিভ GND পিনের সংযোগ দেয়া হয়। পরবর্তী ছবিতে আমরা দেখে নিই কিভাবে এই ডিসপ্লে মডিউলের সাথে JRC বোর্ডের সাধারণ সংযোগ দেয়া যায়।



এখানে তুমি দেখতে পাচ্ছো যে এখানে ডিসপ্লে মডিউলে প্রথম দুটি পিন এর সাথে JRC বোর্ডের পাওয়ার পিনের সংযোগ দেয়া হয়েছে। JRC বোর্ড থেকে ডিসপ্লে মডিউলে লেখা প্রদর্শন করতে চাইলে সেক্ষেত্রে যে সকল পিন ব্যবহারের প্রয়োজন হয়না সেটা আমরা আগেই জেনে এসেছি। এখানে RS পিন, EN পিন এবং শেষের চারটি ডাটা লাইন (D4, D5, D6, D7) এর সাথে JRC বোর্ডের ডিজিটাল পিনের সংযোগ দিয়েছি। এই পিনগুলো যেকোন ডিজিটাল পিনে ব্যবহার করতে হবে যায় যেহেতু এখানে ডিজিটাল সিগ্নাল এর মাধ্যমে লেখার ক্রমান্ত প্রদর্শন করতে হচ্ছে। RW পিনের সাথে গ্রাউন্ড এর সংযোগ রাখা হচ্ছে রিডিং মুডের জন্য। এবং VO পিনে আমরা একটি পটেনশিওমিটার থেকে মাঝের পিনের সংযোগ দিচ্ছি যেখানে পটেনশিওমিটার এর দুই প্রান্তের সংযোগ যথাক্রমে 5V এবং গ্রাউন্ডের সংযোগ দেয়া হচ্ছে। এর মানে আমরা পটেনশিওমিটার এর মাধ্যমে এতে 0 থেকে 5 ভোল্টের মধ্যে যেকোন ভোল্টেজ নিয়ন্ত্রিত করায় দিচ্ছি এবং দরকারমতো ভোল্টেজ দিয়ে কন্ট্রাস্ট এডজাস্ট করে নিচ্ছি। কন্ট্রাস্ট ক্রিকম থাকা উচিত সে ব্যাপারে নিচে ছবি দিয়ে ব্যাখ্যা করার চেষ্টা করা হচ্ছে:



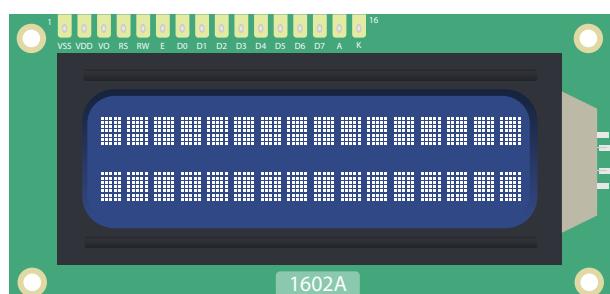
কন্ট্রাস্ট কম



কন্ট্রাস্ট পারফেক্ট

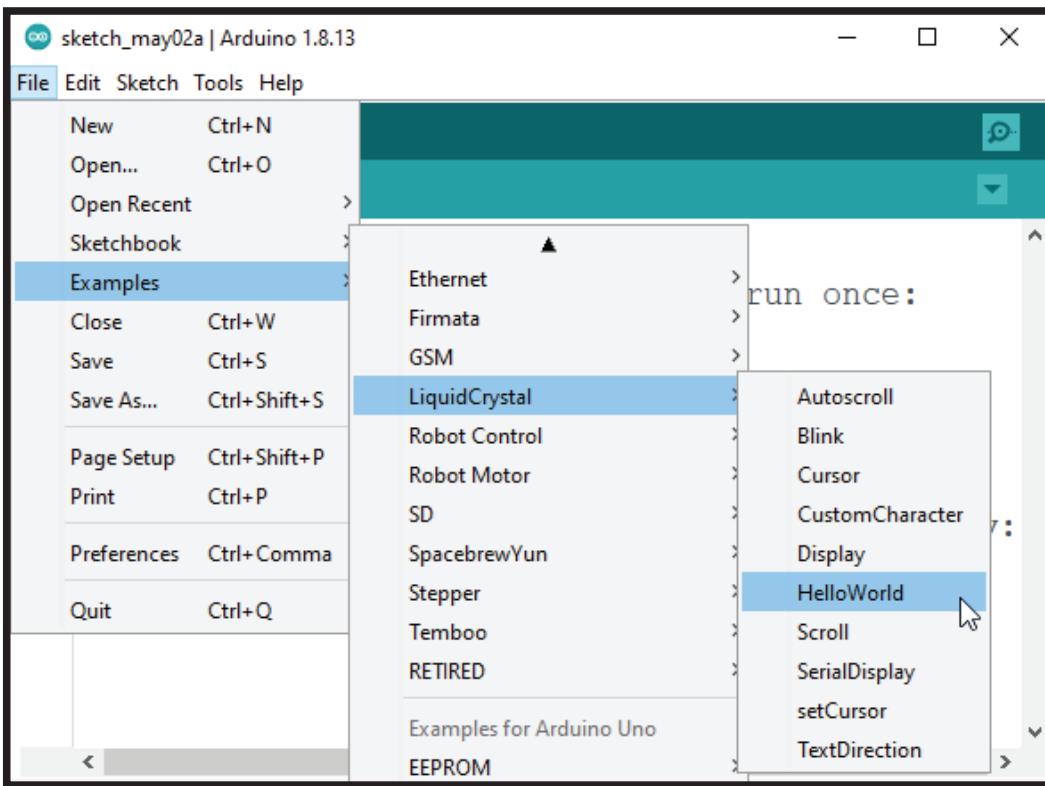


কন্ট্রাস্ট একটু বেশি



কন্ট্রাস্ট অতিরিক্ত

ছবি দেখে তুমি নিশ্চয়ই বুঝতে পারছো যে কন্ট্রাস্ট লেভেল ক্রিকম থাকা উচিত। এই কন্ট্রাস্ট লেভেলে আনতে পটেনশিওমিটার কে স্থানিয়ে ফিরিয়ে এডজাস্ট করতে হবে। যদিও আমরা এখনো লেখা প্রদর্শন করার কোড দেইনি, তবে কোড দিলে যদি এইরকম পরিস্থিতি আসে সেক্ষেত্রে কি করতে হবে সেটা নিশ্চয়ই বুঝে গেসো। এর কানেকশন দেখে যতটা কঠিন মনে হয়, এর কোডিং ততোটাই সোজা। কারণ একে কোড করার জন্য বিল্ট ইন লাইব্রেরী আছে আরডুইনো তে। চলো আমরা এক্সাম্পল কোড বের করার প্রসেস দেখে নিই:



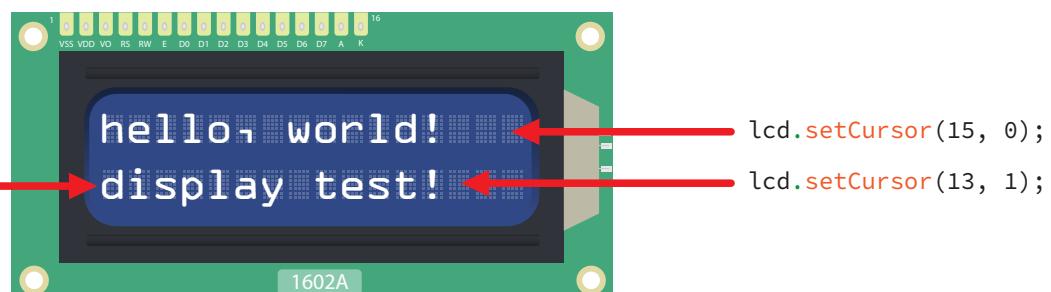
তুমি কি জানো LCD এর পূর্ণ মানে কি? Liquid Crystal Display। এই নামে একটা লাইব্রেরী Arduino IDE তে আগে থেকেই আছে যেটি তোমরা পূর্বে বর্ণিত ছবির ন্যায় খুজে বের করে এক্সাম্পল কোড ওপেন করতে পারো। এই কোডের মূল অংশ নিচে তুলে ধরা হলোঃ

```
#include <LiquidCrystal.h>
int rs=14, en=13, d4=5, d5=23, d6=19, d7=18;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

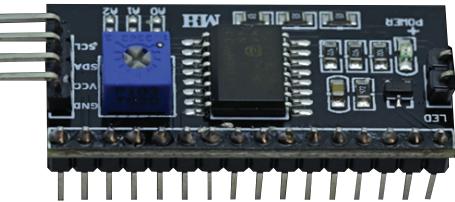
void setup() {
    lcd.begin(16, 2);
    lcd.print("hello, world!");
    lcd.setCursor(0,1);
    lcd.print("display test!");
}

void loop() {
```

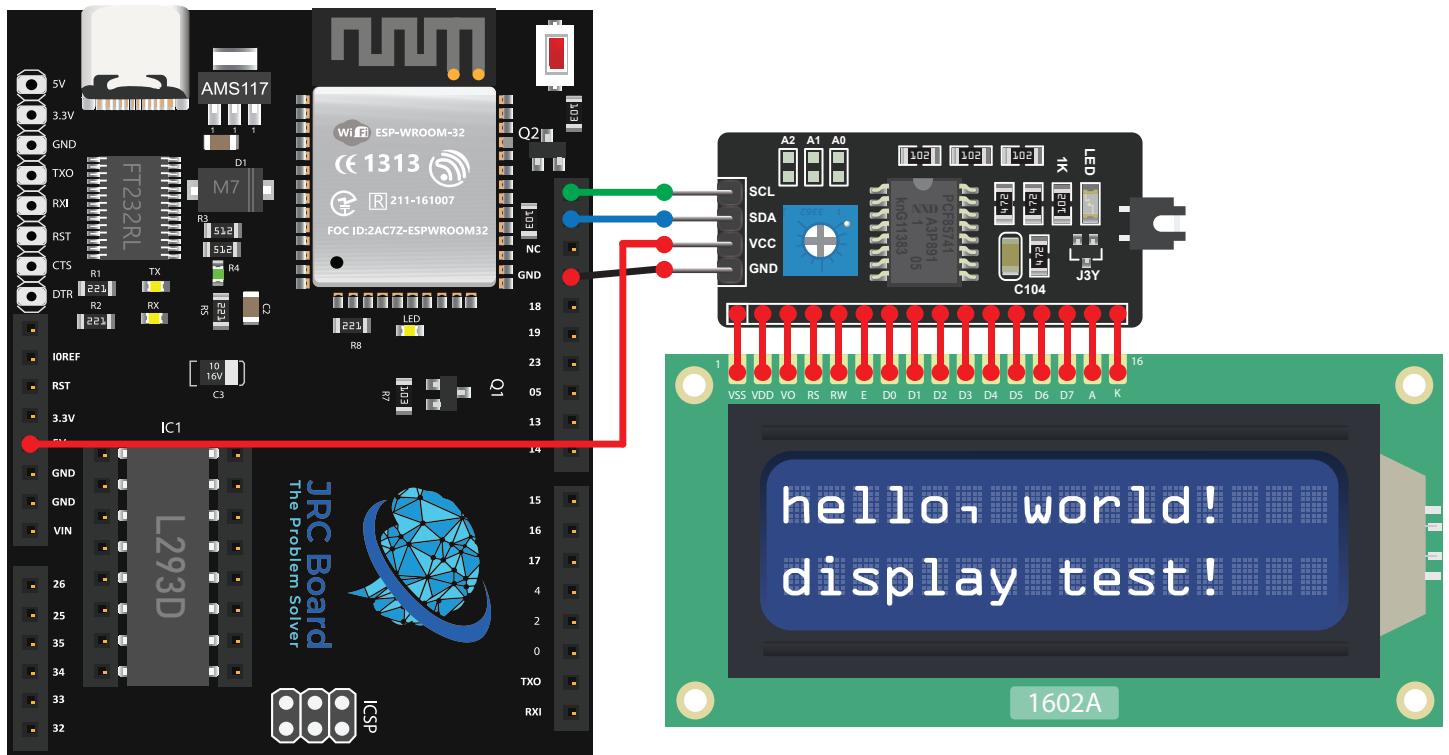
ইতিমধ্যেই আমরা কিভাবে একটা লাইব্রেরী কোডে ইনকুড করতে হয় সেটা জেনে গেছি। এখানে লাইব্রেরীর নাম হলো "LiquidCrystal"। এর ভেতরে তো নতুন একটা ফাংশন তৈরী করতে হবে যার নাম দিয়েছি আমরা `Icd()`। এর ভেতরে আমরা ডিসপ্লে মডিউলের বিভিন্ন পিন JRC বোর্ডের যেসকল পিনের সাথে লাগিয়েছি তার কিন্তু বুবিয়ে দেয়া লাগবে তাইনা? ফাংশনের ভেতরে পিনের নামকরণ দেখলেই বুঝতে পারবে যে ক্রমাগত কোন কোন পিনের নাম্বার বসাতে হবে। এরপর আসি ভেতরে এই ফাংশনের কি কি প্রয়োগ আছে। সত্যিকার অর্থে এর ভেতরে প্রচুর ফাংশন রয়েছে যার সবটুকু কভার না করে আমরা এখানে বেসিক প্রয়োগ টা দেখাচ্ছি। বাকিগুলো তুমি আগের মতো বিভিন্ন এক্সাম্পল কোডে পেয়ে যাবে। শুরুতেই `setup()` ফাংশনের ভেতরে `Icd.begin(16,2)` লিখে ডিসপ্লের সাথে যোগাযোগ চালু করা হয়েছে। এখানে ভেতরে এক সারিতে কয়টা অক্ষর জায়গা নিবে এবং কয়টা সারি রয়েছে সেটা লিখে দেয়া লাগে। যেহেতু আমরা এক সারিতে ১৬ টি অক্ষর পাই এবং মোট ২ সারি রয়েছে, তাই আমরা এখানে (16,2) লিখেছি। যদি তুমি আরও বড়ো সাইজের ডিসপ্লে ব্যবহার করো, সেক্ষেত্রে এই মান পরিবর্তন করতে হবে দরকারমতো। এছাড়াও আরও দুটি ফাংশনের ব্যবহার দেখিয়েছি। একটি হলো `Icd.print()` এবং অপরটি হলো `Icd.setCursor()`। `Icd.setCursor()` দিয়ে কোথা থেকে লেখা শুরু করতে হবে সেটা নির্দেশনা দিয়ে দেয়া যায়। যেমন আমরা যদি এর ভেতরে `Icd.setCursor(0,1)` লিখি, সেক্ষেত্রে দক্ষম প্রথম ঘর এবং দ্বিতীয় সারি থেকে শুরু হবে। উল্লেখ্য যে প্রোগ্রামিং এ গোনা শুরু করা হয় ০ থেকে, সে হিসেবে ঘর এর নির্দেশক নাম্বার হবে ০, ১, ২... এভাবে ১৫ পর্যন্ত। আর সারির নির্দেশন নাম্বার হবে ০ ও ১ এইক্ষেত্রে (১৬*২ ডিসপ্লে ব্যবহার করলে)। এরপর `Icd.print()` ফাংশনের ভেতরে মনমতো কথা লিখে দিলেই এটি ডিসপ্লে মডিউল প্রিন্ট করে দেখায়। যদি আমরা `Icd.setCursor()` ব্যবহার না করি, তবে সর্বশেষ যেখানে লেখা শেষ হয়েছিলো সেখান থেকে লেখা শুরু হবে। উপরের কোডের জন্য এরকম আউটপুট দেখাবেঃ



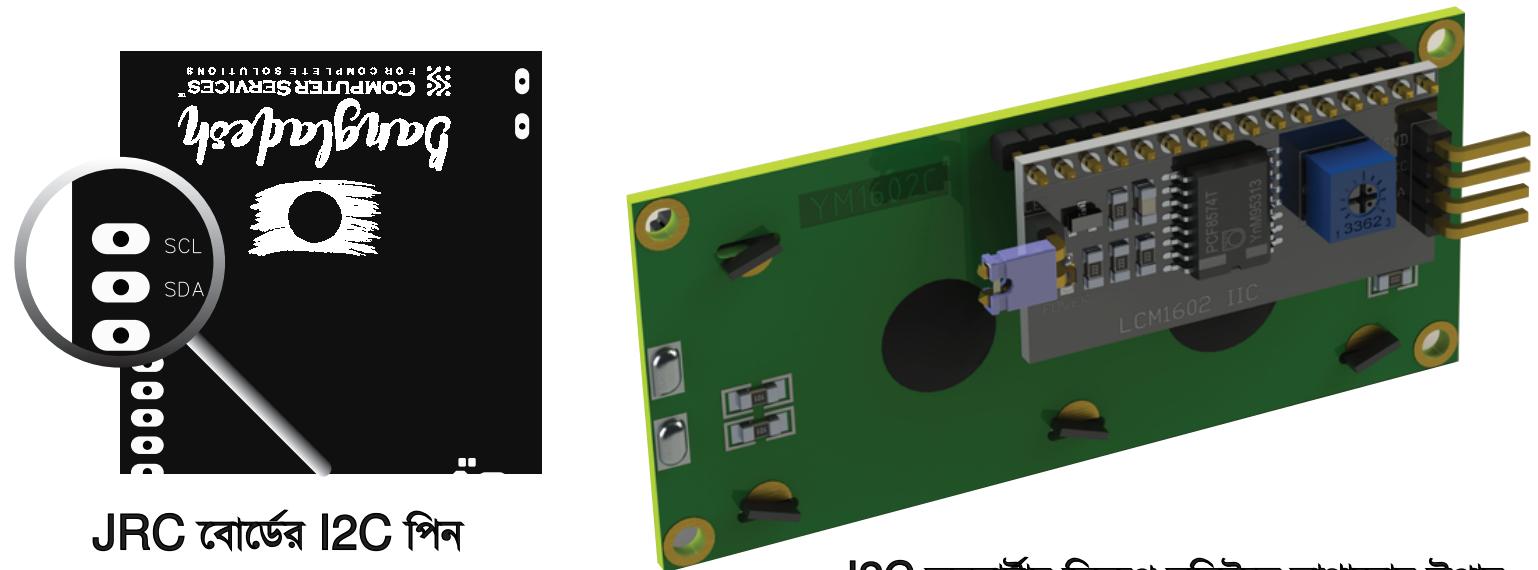
এভাবেই কিন্তু আমরা চাইলে ডিসপ্লে তে যেকোন লেখা দেখাতে পারি। কিন্তু এখানে একটা ছোট সমস্যা হলো আমাদের বেশি সংখ্যক পিনের সংযোগ দিতে হয় যার ফলে তারের একটা জঙ্গল দেখা যায়। এর সমাধান হিসেবে আমরা LCD ডিসপ্লে এর সাথে একটা I2C কনভার্টার ব্যবহার করতে পারি যা দেখতে এরকমঃ



এখানে দেখতে পাচ্ছো যে এই মডিউলের এক পাশে 8 টি পিন রয়েছে এবং অপর দিকে 16 টি পিন রয়েছে। এই 16 টি সরাসরি LCD ডিসপ্লে মডিউলের সাথে সংযোগ স্থাপন করতে হয়। এটি চাইলে মডিউলের সাথে সরাসরি ঝালাই করে দেয়া যায় যার ফলে জিনিস টি আরও নিরাপদ থাকে। নিচে দেখানো হচ্ছে কিভাবে সংযোগ দেয়া যায়ঃ



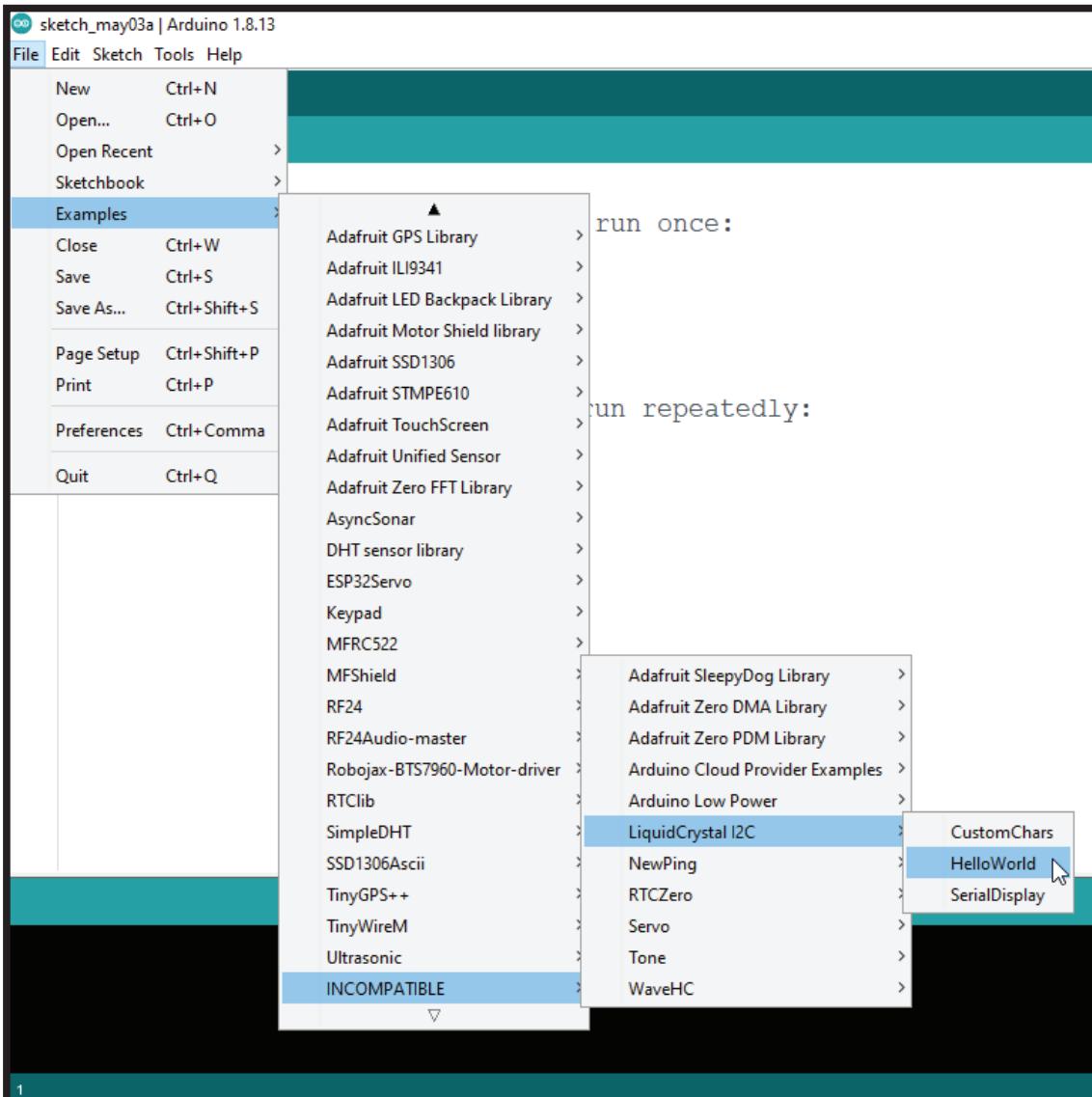
এখানে লক্ষ্য করলে দেখতে পাবে যে ডিসপ্লে মডিউল থেকে যে 16 টা কানেকশন বের হতো সেটা এই কনভার্টার এর মাধ্যমে কমিয়ে এনে মাত্র 8টি কানেকশনের মাধ্যমে JRC বোর্ডের সাথে যোগাযোগ এর কাজ টি করে যাচ্ছে। এর মধ্যে 2টি হচ্ছে পাওয়ার দেবার জন্য (VCC এবং GND) এবং বাকি দুটি হচ্ছে SCL এবং SDA পিন। এগুলো মূলত I2C কমিউনিকেশন (Inter-Integrated Circuit) এর মাধ্যমে ডেটা আদান প্রদানের কাজ করে থাকে। এখানে SDA বা Serial Data পিনটিই মূলত ডাটা পাঠানোর কাজ করে থাকে এবং SCL বা Serial Clock পিনের কাজ হলো দুই ডিভাইসের মধ্যে সময়ের হিসেব ঠিক রাখা যাতে করে ডেটা রিডিং এ কোনপ্রকার ভুল না হয়। JRC বোর্ডে কিন্তু I2C কমিউনিকেশন এর জন্য SCL ও SDA পিন রয়েছে যা কিনা একেবারে উপরে ডান কোণার দুটি পিন এ অবস্থিত। এর মার্কিং বোর্ডের পিছনে লিখে দেয়া আছে। আমরা JRC বোর্ডের SCL পিনের সাথে কনভার্টার এর SCL, এবং বোর্ডের SDA পিনের সাথে কনভার্টার এর SDA পিনের সংযোগ দিবো যা চিত্রে দেখানো আছে।



JRC বোর্ডের I2C পিন

I2C কনভার্টার ডিসপ্লে মডিউলে লাগানোর উপায়

যেহেতু এখানে I2C কমিউনিকেশন এর মাধ্যমে ডিসপ্লে তে লেখা প্রদর্শনের কাজ করতে হবে, তাই এখানে কোডের সামান্য পরিবর্তন আসবে। প্রথমত এটার লাইব্রেরী আগের তুলনায় একটু আলাদা। এটার ক্ষেত্রে এক্সাম্পল কোড বের করার সিস্টেম দেখিয়ে দেয়া হচ্ছে:



দেখতেই পাচ্ছা যে এটা "incompatible" এর তালিকা থেকে বের করতে হচ্ছে। কিন্তু বাস্তব ক্ষেত্রে এটি JRC বোর্ডের সাথে খুব সুন্দরমতো কাজ করবে। আগে আমরা ডিসপ্লে তে যে লেখা দেখিয়েছি সেই লেখাটিই যদি এই সিস্টেমে দেখাতে চাই, সেক্ষেত্রে নিচের মতো কোড করা যায়:

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

void setup() {
  lcd.init(); lcd.backlight();
  lcd.print("hello, world!");
  lcd.setCursor(0,1);
  lcd.print("display test!");
}

void loop() {
}
```

এখানে দেখতে পাচ্ছা শুরুতেই `#include <Wire.h>` লেখা হয়েছে এই I2C কমিউনিকেশন টা কাজে লাগানোর জন্য। এবং লাইব্রেরীর নাম `LiquidCrystal` এর বদলে `LiquidCrystal_I2C` ব্যবহার করে হয়েছে। মজার কথা হলো এখানে আর পিন নাম্বার লেখার ঝামেলা নেই। যখন `lcd` নামে ফাংশন কল করা হচ্ছে, তখন ভেতরে শুরুতে একটা এড্রেস বা ঠিকানা লিখতে হয়। I2C কমিউনিকেশনে একই পিনে অনেকগুলো ডিভাইস লাগিয়ে রেখে নিজেদের মধ্যে যোগাযোগ করা যায়। ব্যাপার টা একটা বাসায় ৫-৬ জন মিলে একটা পরিবারের মতো। তো সেখানে একজনের সাথে আলাদা করে কথা বলার উপায় কি? তার নাম ধরে ডাকবে তাইনা? এইরকমই এই I2C কনভার্টার এর সাথে কথা বলতে গেলে 0x27 এড্রেসে তার সাথে কথা বলতে হবে। এখানে "0x" দিয়ে হেক্সাডেসিমাল সংখ্যা নির্দেশ করা হয়। 0x27 মানে হেক্সাডেসিমাল নাম্বার 27 যার বাইনারী এড্রেস হলো 00100111। এই এড্রেস লেখার পরে ডিসপ্লে মডিউলের সাইজ লিখে দিচ্ছি যেখানে আমাদের 16*2 সাইজের ডিসপ্লের জন্য লিখতে হয়েছে `lcd(0x27,16,2);`। এরপর `setup()` ফাংশনের ভেতরে আমরা আগে `lcd.begin()` দিয়ে কথা বলা শুরু করতাম, এখন সেটার বদলে `lcd.init()` দিয়ে কথা বলা শুরু করতে হবে। যদি আমরা চাই যে এলসিডি ডিসপ্লের ব্যাকলাইট জ্বলে থাকুক, সেক্ষেত্রে `lcd.backlight();` লিখে দেয়া লাগবে। বাদবাকি কোড একই থাকবে।