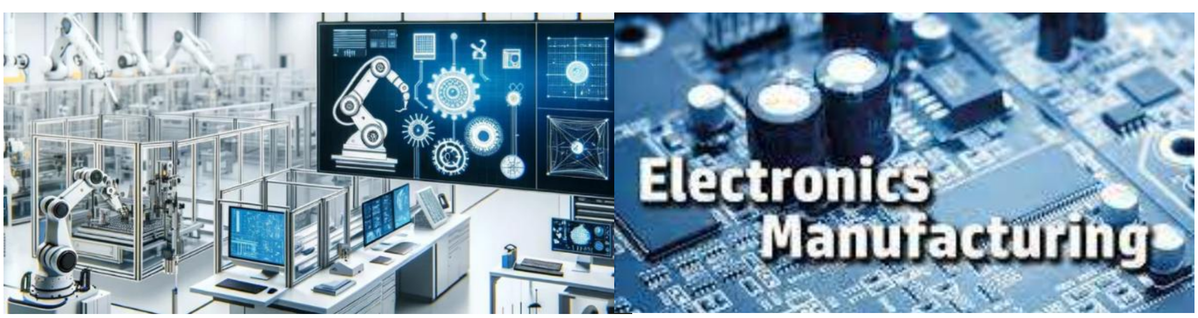
Contents

Energy Systems Production



https://www.scitechto.com/what-is-advanced-manufacturing-technology/

The DeSIRE Advanced Manufacturing course defines and uses the term "Energy Systems Production" as using the latest in technologies, processes, operations, product quality, and cost efficiency to produce energy consuming technologies (e.g., electronics, electrical devices, robotics, sensors for automation processes), systems (e.g., computing hardware, software applications, coding methods), and engineering design process methods needed to manufacture product(s) requiring an energy source to safely operate in the classroom (i.e., electricity, chemical, hydropower, solar).

By a definition, energy systems manufacturing involves the production of equipment and systems that store, generate, and distribute energy.

Across manufacturing and advanced manufacturing industries, energy systems are the most commonly used technologies, namely machinery and robotics. Reason being, these technologies are essential to driving lower production and labor costs, including product efficiency and quality. The DeSIRE lesson plans and hands-on project-based learning for food, pharmaceutical, and energy systems mock advanced manufacturing simulations should design and build robotics and automation process applications to showcase their value in industry. Robots are used in mobile platforms, articulated arms, and related automation technologies in production assembly and test processes.

Engineers design specific applications for robots to increase production and process efficiency. Engineering design processes are used to constantly improve tasks and production requirements to reduce material costs, waste, and labor expenses.

History of Manufacturing

Time Required: 30 Minutes

Lesson summary

Many food, pharmaceutical, and energy system technology products are made through the process of manufacturing. Take a look at the history of manufacturing and conduct some research of your own! Through your research, learn about the history of manufacturing and develop your skills using technology. Think critically about the importance of manufacturing to our town/city locally.

Introductory activity - Guiding Questions

- What is manufacturing?
- How were items created in the past vs the present?
- What are some products that are manufactured?
- What were some of the major developments made in manufacturing?
- Provide some examples of the technology that has been created that benefits/is used in manufacturing
- How has manufacturing developed over time?
- How were products produced in the past (when modern technology was not available)?
- How is robotics included in manufacturing?

Learning activities

Step 1: Introduce the lesson by watching manufacturing videos and discussing the guiding questions

Step 2: Have students work either individually or in groups to research the guiding questions

Step 3: Students will gather information about manufacturing

Step 4: Create a discussion board (either on the smartboard or on a piece of poster paper) with the information students gathered

Step 5: Create a timeline of how manufacturing has developed

Step 6: Culminating Activity

Culminating activity - students will:

- Journal their discoveries
- Revisit the guiding questions
- Journal any remaining questions they have about the activity and manufacturing

Sources resources:

- https://pbsnc.pbslearningmedia.org/
- https://www.teachengineering.org/

Introduction to Advanced Manufacturing

Time Requirement: 40 Minutes

Lesson Summary:

By the end of this lesson students will be able to understand what assembly lines are, how they impact the manufacturing of goods, and how humans on the assembly line versus robotics on the assembly line will impact the production and quality of goods.

Introductory Activity:

- What do we know about assembly lines?
- Show video clips of assembly lines in TV shows
    - Suggestions: I Love Lucy Chocolate Assembly Line scene
    - Drake and Josh Sushi Assembly Line scene

Learning Activities

- Artisan versus Factory
    - Have each individual student cut a snowflake out of coffee filters into any pattern they wish in 5 minutes
        - Discuss how many were made and how they differ

  - Put students into groups to act as a factory. Each student will have one role and a repetitive role to create a snowflake. The goal is to make the same product. How many can be produced in 5 minutes
    - Discuss how many were made and how they differ versus the goods produced by individual artisans
  - Repeat trials of factory scenario to see how to improve the line
  - What are the pros and cons of artisans versus factory workers
- Factory: Humans versus Robotics
  - Show video clips of robotic assembly lines
  - Discuss how an assembly line of humans and robots will differ in production, quality of goods, and resources needed to produce
    - Extension: discuss the economic, social, and educational impacts of switching from human to robotic assembly lines

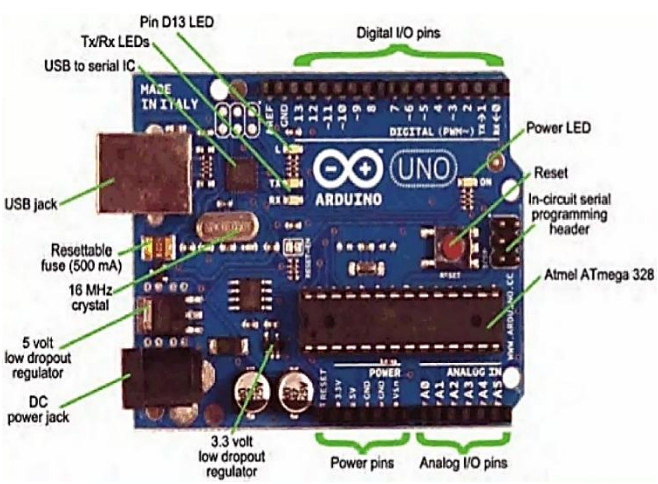
Oral and Written Communication

The DeSIRE Advanced Manufacturing course defines and uses the term "Oral and Written Communication" as a way for students to learn, develop, and enhance verbal and written communication skills. The learning outcomes are to develop skills and abilities to articulate thoughts and express ideas effectively using oral, written and non-verbal communication skills (to inform, instruct, and persuade). The communications include effective listening for meaning and understanding.

DeSIRE student learners should engage in a wide range of oral and written communications projects (i.e., written technical reports and papers, slide presentations, technical demonstrations, STEM fair, Careers Fair, interviews with professionals, shadow experiences) to develop critical thinking, and team-based collaboration.

What is an Arduino microcontroller?

Arduino is an open-source circuit board and computing platform used for building and controlling electronics projects. There are many models of the Arduino brand. The DeSIRE program uses the UNO model. The Arduino consists of both a physical programmable circuit board (also known as a microcontroller) and software (IDE - Integrated Development Environment) that runs on your computer. The IDE loaded on your computer is used to write and upload computer code to the physical Arduino circuit board.



https://learn.sparkfun.com/tutorials/what-is-an-arduino/the-arduino-family

STEM Inventory for Lesson Plans and PBL Activities

Introduction to Google Sheets

Time Requirement: 60 Minutes

Lesson Summary:
Students will be able to enter data, tabulate data, learn a few shortcuts, and create graphs/tables from data entered. Students will complete a scavenger hunt worksheet that requires them to follow steps for data entry, tabulation, and graph creation in Google Sheets. The final product will be a line graph based on the data they enter.

Introductory Activity:

- Begin with a brief discussion on how data is used in everyday life (e.g., budgets, sports statistics).
- Ask students, "How do you think data can help us visualize information?"
    - Introduce a fun fact about data visualization to spark interest
- Demonstrate the Google Sheets interface, highlighting key features.
    - Show how to enter data into cells and format it (e.g., bold, color).
    - Introduce shortcuts (e.g., copy, paste, undo).
- Explain how to select data for graph creation.
    - Common Misconception: Students may think that graph creation is only for specific types of data; clarify that any data can be visualized.

Learning Activities:

Guided Practice:

- In pairs, students will practice entering sample data provided by the teacher.
- Monitor students as they work, asking guiding questions like:
    - "What happens when you change the data in this cell?"
    - "Can you show me how to format this data?"
- Offer support for students struggling with shortcuts or data entry.

Independent Practice:

- Students will receive a scavenger hunt worksheet with specific tasks to complete in Google Sheets, including:
    - Entering a specified set of data.

- Creating a line graph based on their data.
- Formatting the graph with labels and colors.

Extension Activity:

Students who finish early can explore additional graph types (e.g., bar graphs, pie charts) and create a new graph from the same data set.

Paper Circuit Starter Kit

Recommended for DeSIRE Grade(s) and Cohort Year(s):

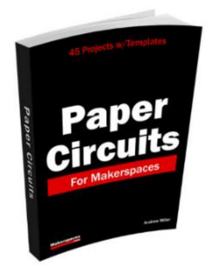
6th, 7th Grades for program years 1 and 2

Description (Overview):

Why used for DeSIRE? This product kit provides students hands-on experiences with 45 Paper Circuit project options, LEDs, low power coin cell batteries, and copper tape for circuit wiring.

How will this be used for DeSIRE? Students will be assigned personal and team-based projects to learn basic electricity and electronics, develop hands-on experiences, build confidence with entry level circuits, and related presentations about STEM (i.e., oral, written, and/or project demonstrations).

Paper Circuits Starter Kit w/ Ebook (EDU Bundle)



Paper Circuits Project Book (Paperback or PDF)

Vendor created Lesson Plans *(Stored in NCSU DeSIRE Google Drive)*: https://drive.google.com/file/d/1wJzpboHwvlI4z0ssUHc0oQk5zRNtjyW3/view?usp=drive_link

Vendor/supplier resource (image source reference)**:** https://www.makerspaces.com/

Special Requirements needed: None.

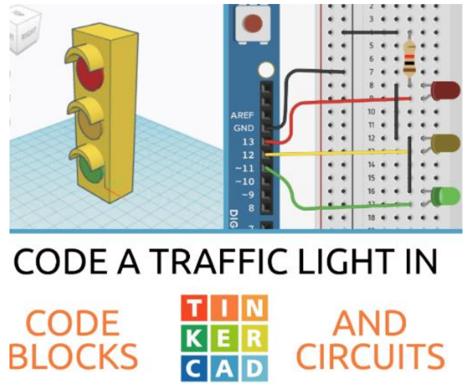
Tinkercad (Circuits, 3D Modeling, and Coding)

Recommended for DeSIRE Grade(s) and Cohort Year(s):

6th, 7th, 8th Grades for program years 1, 2, 3, and 4

Description (Overview):

Why used for DeSIRE? Tinkercad is a free, online 3D modeling program that is used for creating 3D designs, printing, electronics, and coding. It is known for being easy to use, with a simple interface that is good for beginners.



How will this be used for DeSIRE? First, students will build upon hands-on experiences gained using Paper Circuits. Following, students will be assigned personal and team-based projects to learn at the basic, intermediate, and advanced levels of applications used across advanced manufacturing (e.g., electricity, electronics, coding, mechanical, 3D design and printing, and present the results (i.e., oral, written, and/or project demonstrations). Advanced applications used in advanced manufacturing will be simulated using the Arduino microcontroller, read and layout circuits, identify circuit components, and ultimately build virtual circuits and systems that will be turned into real physical products. Students will conduct related presentations about STEM (i.e., oral, written, and/or project demonstrations).

Vendor created Lesson Plans (Online): https://www.tinkercad.com/

Vendor/supplier resource (image reference source): https://www.tinkercad.com/

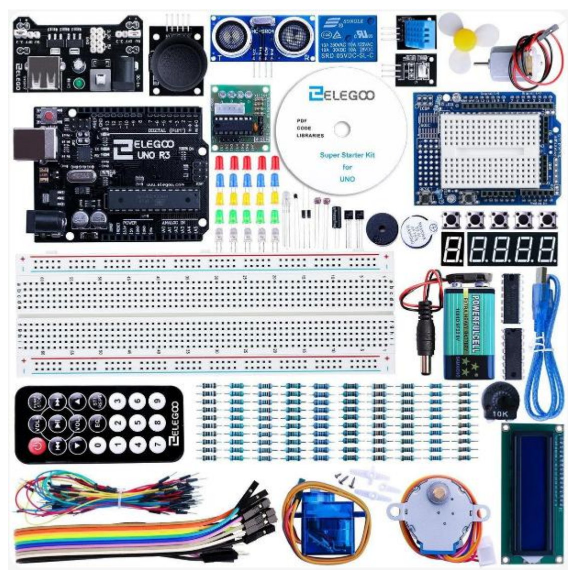
Special Requirements needed: Yes. Teachers and students must create user accounts unless the teacher setup access codes for each student. See Tinkercad for more details.

Elegoo Arduino Uno R3 Project Kit (200+ Components)

Recommended for DeSIRE Grade(s) and Cohort Year(s):

6th, 7th, 8th Grades for program years 1, 2, 3, and 4

Description (Overview):



Why used for DeSIRE? The Arduino kit is extremely versatile and comes with over 200+ real-world electronics components, 100+ lesson plans and variations of plans to build circuits and coding used in advanced manufacturing environments around the world. Projects use a wide range of sensors (light, touch, temperature, humidity, distance, etc.), circuits like IC's, transistors, resistors, switches, wires, etc. Real engineers use these kits.

How will this be used for DeSIRE? First, students will build upon hands-on experiences gained and learned using Paper Circuits and Tinkercad. Students will be assigned personal and team-based projects to learn at the basic, intermediate, and advanced levels of applications used across advanced manufacturing (e.g., electricity, electronics, mechanical, coding, mechanical, and present the results (i.e., oral, written, and/or project demonstrations). Advanced applications used in advanced manufacturing can be completed for mock advanced manufacturing environments. Students will conduct related presentations about STEM (i.e., oral, written, and/or project demonstrations).

Vendor created Lesson Plans *(Stored in NCSU DeSIRE Google Drive):* https://drive.google.com/file/d/1cEjM7va22xnEEBFc9h8VLSXPo5f6XpQ4/view?usp=drive_link

Vendor/supplier resource (image resource reference): https://www.elegoo.com/

Special Requirements needed: None.

Code for DHT Sensor, sending Temp Reading to LCD screen, and lighting RGB LED light to signify temperature

https://toptechboy.com/arduino-tutorial-51-dht11-temperature-and-humidity-sensor-with-lcd-display/

```
#include "DHT.h"
#define Type DHT11
#include <LiquidCrystal.h>

// LCD pins
int rs = 7;
int en = 8;
int d4 = 9;
int d5 = 10;
int d6 = 11;
int d7 = 12;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

// DHT11 sensor
int sensePin = 2;
DHT HT(sensePin, Type);
float humidity;
float tempC;
float tempF;
int setTime = 500;
```

```
int dt = 1000;

// RGB LED pins (common cathode)
int redPin = 6;
int greenPin = 5;
int bluePin = 3;

void setup() {
 Serial.begin(9600);
 HT.begin();
 delay(setTime);
 lcd.begin(16, 2);

 // Set RGB LED pins as output
 pinMode(redPin, OUTPUT);
 pinMode(greenPin, OUTPUT);
 pinMode(bluePin, OUTPUT);

 // Turn off LED initially
 digitalWrite(redPin, LOW);
 digitalWrite(greenPin, LOW);
 digitalWrite(bluePin, LOW);
}

void loop() {
 humidity = HT.readHumidity();
 tempC = HT.readTemperature();
 tempF = HT.readTemperature(true);

 // Display on LCD
 lcd.setCursor(0, 0);
 lcd.print("Temp F= ");
```

```
  lcd.print(tempF);
  lcd.setCursor(0, 1);
  lcd.print("Humidity= ");
  lcd.print(humidity);
  lcd.print(" %");
  delay(500);
  lcd.clear();

  // Display on Serial Monitor
  Serial.print("Humidity: ");
  Serial.print(humidity);
  Serial.print("% Temperature ");
  Serial.print(tempC);
  Serial.print(" C ");
  Serial.print(tempF);
  Serial.println(" F ");

  // Change RGB LED color based on temperature
  if (tempF < 75) {
    // Blue
    digitalWrite(redPin, LOW);
    digitalWrite(greenPin, LOW);
    digitalWrite(bluePin, HIGH);
  } else {
    // Red
    digitalWrite(redPin, HIGH);
    digitalWrite(greenPin, LOW);
    digitalWrite(bluePin, LOW);
  }
}
```

xArm Robotic Arm with Arduino Secondary Development Sensor Kit

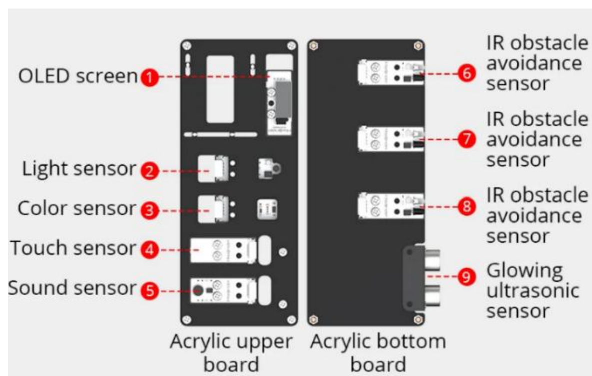
Recommended for DeSIRE Grade(s) and Cohort Year(s):

6th, 7th, 8th Grades for program years 1, 2, 3, 4

Description (Overview):

Why used for DeSIRE? Students need hands-on experiences to understand the value of robotic arms across advanced manufacturing industries: (1) Safety (perform dangerous, heavy, or repetitive tasks), (2) Precision (perform tasks with more consistency, extreme precision and accuracy), (3) Efficiency (robotic arms can operate 24 hours a day, seven days a week, allowing businesses to keep production going continuously. (4) Cost savings (can automate labor-intensive and repetitive tasks, which can significantly reduce production time and labor costs), and (5) Versatility (can be used for a variety of tasks, such as product assembly, wiring, welding, painting, material handling, and pick-and-place tasks).

How will this be used for DeSIRE? Students will use xArm developed lesson plans and DeSIRE teacher enhancements to perform different PBL activities based on advanced manufacturing processes, operations, and environments (e.g., pick and place tasks, precision operations, perform time and cost saving methods in mock assembly operations). Students will conduct related presentations about STEM (i.e., oral, written, and/or project demonstrations).



Vendor created Lesson Plans *(Stored in NCSU DeSIRE Google Drive):* https://drive.google.com/drive/folders/136h-OJSTmhnkv0SiuBUd3UlhK0EfeJ26?usp=drive_link

Vendor/supplier resource (image resource reference): https://www.hiwonder.com/

Special Requirements needed: Yes. Software loaded on computing device (laptop, tablet, phone, desktop). Optional attachment of the development sensor board kit.

Elegoo UNO Arduino R3 Smart Robot Car Kit V4

Recommended for DeSIRE Grade(s) and Cohort Year(s):

6th, 7th, 8th Grades for program years 1, 2, 3, 4

Description (Overview):

Why used for DeSIRE? Students need hands-on experiences to understand the value of robotics usage and applications across advanced manufacturing industries: (1) Efficiency (can work continuously longer than human), (2) Quality (can help ensure consistency and quality by reducing the defects and margin of error), (3) Safety (can handle chemicals and dangerous tasks, lift and move heavy objects, work with blades or lasers), (4) Cost (can reduce labor & transportation expenses), (5) Flexibility (can be programmed to do repeatedly tasks) (6) Collaboration (can work alongside humans), (7) Data generation (can generate large amounts of data and help identify bottlenecks and monitor performance on assembly lines).

How will this be used for DeSIRE? Students will use Elegoo developed lesson plans and DeSIRE teacher enhancements to perform different PBL activities based on advanced manufacturing processes, operations, and environments (e.g., perform transportation destination tasks, precision operations, perform time and cost saving methods in mock assembly operations) across all three DeSIRE focus industries – Energy, Food, and Pharma. Students will conduct related presentations about STEM (i.e., oral, written, and/or project demonstrations).

Vendor created Lesson Plans *(hardcopy manual and online):*
https://us.elegoo.com/pages/download

Vendor/supplier resource (image resource reference): https://www.elegoo.com/

Special Requirements needed: Yes. Software loaded on computing device (laptop, tablet, phone, desktop). Optional attachment of the development sensor board kit.

Ultrasonic Devices

Time Required: 90 Minutes

Summary

This lesson focuses on ultrasound wavelengths and how sound frequencies are used by engineers to help with detection of specific distances to or in materials. Students gain an understanding about how ultrasonic waves are reflected and refracted. Students also see how ultrasound technology is used in medical devices. The activity following this lesson allows students to test their knowledge by using the Elegoo Ultrasonic sensor and Arduino UNO Microcontroller kit.

Advanced Manufacturing Connection

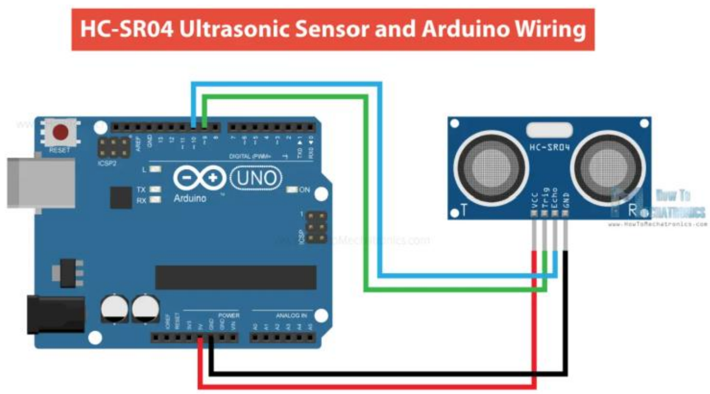
Electrical, mechanical, and computer engineers often use ultrasonic devices to measure and evaluate materials used in manufacturing and building to identify potential defects without destroying the product. This is a much safer and efficient approach than older scanning methods.

After this lesson, students should be able to:

- Identify a way ultrasound is used to help us in everyday life
- Explain why there are different ways to measure with ultrasonic devices
- Explain how an ultrasonic distance sensor measures distance

Lesson Background and Concepts for Teachers

This lesson is designed to engage students in an understanding of how ultrasonic sound is useful to us. It explains what ultrasonic devices are and gives examples of its uses. It also specifically shows the Elegoo HC-SR04 ultrasonic distance sensor. This lesson is connected to the associated activity Designing and Packaging a Distance-Sensing Product - where students use web editing software with the Elegoo HC-SR04 distance sensor and Arduino Microcontroller, Designing and Packaging a Distance Sensing Product.



https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/

Reference Elegoo documents**:**

- Basic Starter Kit for UNO V1.0.2019.07.24.pdf
- https://www.teachengineering.org/lessons/view/mis-2227-ultrasonics-uses-arduino-ultrasound-technology
- https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/

xArm Robot

- **Lesson Plan documents below are *linked to DeSIRE Google Drive:***

  1. PC Software Instruction
  2. xArm 1S Robotic Arm
  3. Bus Servo Learning

4. Action Programming
5. Integrate Action Files
6. Offline Running
7. Action Group Download

Hiwonder Software Downloads (PC) **"xArm Desktop, Libraries, Action Groups"**

- https://drive.google.com/drive/folders/1_QGlHXSKEcr_5BcYU1lXJWEpH9ceoT9H

- **Hiwonder Software Downloads** (Mac)
  - Use the Apple App Store – Hiwonder XArm
- **Hiwonder Software Downloads** (PC) "xArm Desktop, Libraries, Action Groups, etc.
  - https://drive.google.com/drive/folders/1UdS4FKLJgCnbv-GaNEQB96wi4HYrfWmd
- Using the robotics arm technology in three main methods of control:
  - Manual recording of arm movement and positioning (numerical control)
  - Wireless PS/2 joystick controller
  - Computer software control
- Lunch (optional working lunch)
- Once proficiency is achieved using the robotic arms, prepare to proceed to arm and sensor board lesson plan activities.
  - **The following documents are linked from the Hiwonder Download site:** https://drive.google.com/drive/folders/1_QGlHXSKEcr_5BcYU1lXJWEpH9ceoT9H
    - 0. Read First
    - 1.Getting Ready
    - Lesson 2 Touch Control

Smart Robot Car V4.0 with camera : The DeSIRE Smart Robot Car V4.0 with camera kits comes with documentation and all parts. The online documentation (Arduino C++ files, assembly) found at: https://www.elegoo.com/pages/download

- Software Applications found on iTunes and Google Play Store (ElegooKit)

- Build and code the FPV (First Person View) robotics control experience and using Wi-Fi for Video transmission and control the robot car's movement.
- Developed from Arduino open source platform. DIY assembly and construction will help to cultivate student's critical thinking concentration and hands-on ability.
- Arduino-based C++ and Graphical programming provides special building block functions, and with a variety of robotics functions used in manufacturing operations.

xArm Robotics (Continued)

- Once proficiency is achieved with the introductory documents and Lesson #2 (Touch Control), please proceed to the following lesson plan activities:

  Note the following documents are linked from the Hiwonder Download site:
  https://drive.google.com/drive/folders/1_QGlHXSKEcr_5BcYU1lXJWEpH9ceoT9H

  Lesson 3 Claw Machine

  Lesson 4 Password Lock

  Lesson 5 Intelligent Sound Control

  Lesson 6 Auto Sorting

  Lesson 7 Distance Measurement

  Lesson 8 Grab at a Certain Distance

  Lesson 9 Light Sensitivity

  Lesson 10 Color Identify

  Lesson 11 Color Sorting

Arduino Temperature & Humidity Sensor Project

Lesson Objective:

• Learn basic electronics by assembling an Arduino-based sensor circuit.

• Write/upload code that reads temperature/humidity from a DHT11 (or similar) sensor, displays the readings on an LCD, and prints them to the Serial Monitor.

• Collect environmental data from different locations around school.

• Organize the collected data into tables, create bar graphs using spreadsheet software, and answer analysis questions regarding observed patterns.

Materials Needed:

• Arduino boards with USB cables

• DHT11 or similar temperature/humidity sensor modules

• LCD display (e.g., 16×2 character LCD)

• Breadboards and jumper wires

• Computer/laptop with the Arduino IDE installed (with Adafruit DHT and LiquidCrystal libraries)

• Spreadsheet software (Microsoft Excel, Google Sheets, etc.)

• Printed copies of a data table template (or a digital sheet)

• Access to a computer lab or individual student computers

Day 1: Introduction & Circuit Assembly (45 minutes)

Time Breakdown:

5 minutes – Introduction and Objective Review

– Explain the goals of the project.

– Introduce basic concepts: sensors, Arduino boards, and real-world applications like weather stations.

– Show examples of an Arduino board, a DHT11 sensor module, and an LCD display.

20 minutes – Demonstration of Circuit Assembly

– On a large demonstration breadboard, show how to:

Connect the power (5V) and ground.

• Attach the data pin(s) from the DHT11 to a digital pin on the Arduino.

– (If using an LCD, also show how to connect it following its wiring diagram.)

– Emphasize careful attention to wiring connections.

15 minutes – Hands-On Circuit Building

– Have each student or pair work on their own breadboard assembly.

– Circulate around the room to offer guidance and troubleshoot any connection issues.

5 minutes – Wrap-Up Discussion

– Recap key points (sensor function, proper wiring).

– Answer any questions regarding circuit construction.

Day 2: Programming & Field Data Collection (45 minutes)

- Time Breakdown:

  • 10 minutes – Review and Preparation

  – Quickly recap Day 1's work.

  – Remind students to check their circuits are correctly assembled.

  – Introduce the Arduino code that reads from the sensor, displays data on an LCD, and prints to the Serial Monitor.

- **15 minutes – Code Walkthrough & Upload**
  – Provide the following code:

Code

```
#include "DHT.h"
#define Type DHT11
#include <LiquidCrystal.h>
int rs=7;
int en=8;
int d4=9;
int d5=10;
int d6=11;
int d7=12;
LiquidCrystal lcd(rs,en,d4,d5,d6,d7);

int sensePin=2;
DHT HT(sensePin,Type);
float humidity;
float tempC;
float tempF;
int setTime=500;
int dt=1000;

void setup() {
 // put your setup code here, to run once:
Serial.begin(9600);
HT.begin();
delay(setTime);
lcd.begin(16,2);
}
```

```
void loop() {
humidity=HT.readHumidity();
tempC=HT.readTemperature();
tempF=HT.readTemperature(true);

lcd.setCursor(0,0);
lcd.print("Temp C= ");
lcd.print(tempF);
lcd.setCursor(0,1);
lcd.print("Humidity= ");
lcd.print(humidity);
lcd.print(" %");
delay(500);
lcd.clear();
Serial.print("Humidity: ");
Serial.print(humidity);
Serial.print("% Temperature ");
Serial.print(tempC);
Serial.print(" C ");
Serial.print(tempF);
Serial.println(" F ");
}
```

– Explain key parts of the code (e.g., sensor reading, LCD commands).

– Verify that there are no errors when compiling in the Arduino IDE.

– Upload the code to each student's Arduino board.

- 15 minutes – Bench Testing & Troubleshooting

  – Instruct students to open the Serial Monitor and verify that data is being printed.

  – Check the LCD display for clear readings.

  – Resolve any issues with wiring or code errors together as a class.

- 5 minutes – Field Data Collection Planning

  – Divide the class into 5-6 groups (each group collects data corresponding to):

  • Gym

  • Cafeteria

  • 7th Grade Hallway

  • 8th Grade Hallway

  • Library

  • Main Hallway

- – Instruct groups to let their Arduino units run for at least 2-3 minutes at their each location.
- – Remind them to note the environment (e.g., "near windows," "crowded space") and record key observations.

Day 3: Data Organization, Graphing & Analysis (45 minutes)

- Time Breakdown:

15 minutes – Organizing Collected Data

– Have each group compile their collected readings into a table.

• Example Table Format:

Location	Temperature (℃)	Humidity (%)
Gym		
Cafeteria		
7th Hallway		
8th Hallway		
Main Hallway		
Library		
Outside (Control)		

– Discuss how to calculate averages from multiple readings.

– Provide printed templates or digital spreadsheet templates for students to record their data.

15 minutes – Creating Bar Graphs

– Guide students in using a spreadsheet tool (Excel/Google Sheets) to input their table data.

– Instruct them on how to:

• Create one bar graph that compares the Temperature across all six locations.

• Create another bar graph comparing the Humidity across the same locations.

– Emphasize labeling axes, giving titles, and using distinct colors for clarity.

10 minutes – Analysis Questions & Discussion

– Once graphs are complete, have students answer follow-up questions:

1. Which location showed the highest temperature? What factors might contribute to this difference?
2. Which location had the lowest humidity? How does its climate compare with the other areas in the building?
3. Based on your bar graphs, what trends or patterns do you observe regarding temperature and humidity variations across locations?
4. Can you think of any sources of error or environmental factors that may have affected your data collection (e.g., direct sunlight, ventilation)?
5. How might this type of sensor network be used in a real-world situation? Provide one example.

- Encourage students to present their findings and graphs along with their answers.
- 5 minutes – Wrap-Up & Reflection
  – Summarize the skills learned (circuit building, programming, data analysis).
  – Highlight how combining hardware projects with data visualization deepens understanding of environmental monitoring.
  – Answer any remaining questions and encourage students to think about future projects or real-world applications.

Engineering Belt Conveyor Systems for Pharmaceutical Manufacturing

- **Grade Level:** Middle School (6–8) **Duration:** 3 Class Periods (Approx. 45–60 min each)
- **STEM Focus:** Engineering Design, Robotics, Automation, Manufacturing Standards Alignment
- **NGSS MS-ETS1-1**: Define the criteria and constraints of a design problem with sufficient precision.
- **NGSS MS-ETS1-4**: Develop a model to generate data for iterative testing and modification.
- **CCSS.ELA-LITERACY.RST.6-8.3**: Follow multi-step procedures precisely.
- **ISTE 5b**: Collect data and represent it visually to emphasize problem-solving.

Learning Objectives: By the end of this lesson, students will be able to:

1. Describe common conveyance methods used in pharmaceutical manufacturing.
2. Explain how belt conveyors move products efficiently in a cleanroom setting.
3. Build and wire a 2-motor Arduino-controlled conveyor system with IR object detection.
4. Modify and upload Arduino code to control the conveyors.
5. Test, troubleshoot, and optimize conveyor speed and object counting functions.
6. Reflect on real-world manufacturing applications of their design.

Day-by-Day Plan

Day 1 — Background Knowledge & Introduction

- **Objective:** Students will learn how materials are moved in pharmaceutical manufacturing and understand the basics of belt conveyors.

Materials:

- Slides/video showing real pharma manufacturing lines
- Handouts on conveyance methods (belt, roller, pneumatic)
- Images/diagrams of cleanroom conveyor setups
- Whiteboard/markers

Activities:

1. **Hook (5 min) –** Show a short video of pill bottles being filled and transported via conveyors in a pharmaceutical plant.
2. **Mini-Lecture (15 min) –** Discuss:
    - Why pharmaceutical manufacturing requires precision and cleanliness.
    - Conveyance types: belt conveyors (today's focus), roller conveyors, pneumatic systems.
    - How sensors help with quality control.

3. **Interactive Discussion (10 min) –** Students brainstorm challenges conveyors might face in a cleanroom.

4. **Preview Build Task (10 min) –** Introduce the project: students will build a 2-conveyor Arduino system with IR object detection that counts pill bottles.

5. **Wrap-Up (5 min) –** Assign a short reading on conveyor belt design in manufacturing.

   Homework: Research one real-world use of belt conveyors in the medical or pharmaceutical industry.

Day 2 — Design & Build

- **Objective:**
  - Students will begin assembling their 2-conveyor system using Arduino, motors, IR sensors, and 7-segment displays.
- **Materials:**

  - Arduino Uno boards
  - Breadboards & jumper wires
  - 2 DC motors + motor driver modules (L298N or similar)
  - IR sensors
  - 7-segment display with 74HC595 shift register
  - Small conveyor belt kits (or custom built)
  - Laptops with Arduino IDE installed
  - Screwdrivers, hot glue, mounting hardware

  Activities:

  - **Safety & Setup (5 min) –** Review safe handling of electronics and mechanical parts.
  - **Review Wiring Plan (10 min) –** Go over the motor driver, IR sensor, and display wiring.
  - **Hands-On Build (30 min) –** Students work in groups to:

    1. Mount motors to conveyor frames
    2. Wire motors to Arduino via motor driver
    3. Wire IR sensor to detect passing objects
       - Wire 74HC595 to 4-digit 7-segment display

Pin Assignments for Arduino to 74HC595 and 4-Digit 7-Segment Display

Arduino → 74HC595

74HC595 Pin	Function	Connects To	Notes
8	GND	Arduino GND	Common ground
16	VCC	Arduino 5V	Power supply for logic
10 (MR)	Master Reset	Arduino 5V	Tie HIGH for normal operation
13 (OE)	Output Enable	Arduino GND	Tie LOW for always enabled
14 (DS)	Data	Arduino Pin 4	Serial data in
12 (STCP)	Latch Clock	Arduino Pin 5	Transfers register to output
11 (SHCP)	Shift Clock	Arduino Pin 6	Shifts data bits in
Q0–Q7 (15,1,2,3,4,5,6,7)	Segment outputs a–g + DP	To 7-seg segment pins (through resistors)	220Ω recommended

74HC595 → 4-Digit 7-Segment Display

74HC595 Output	Segment
Q0	Segment A
Q1	Segment B
Q2	Segment C
Q3	Segment D
Q4	Segment E
Q5	Segment F
Q6	Segment G
Q7	Decimal Point (DP)

Each segment line goes through a 220Ω resistor to the display pin.

Digit Control (Multiplexing)

Digit	Arduino Pin	Transistor Base	Display Pin
D1 (leftmost)	Pin 11	via 1kΩ resistor	Digit common 1
D2	Pin 10	via 1kΩ resistor	Digit common 2
D3	Pin 12	via 1kΩ resistor	Digit common 3
D4 (rightmost)	Pin 8	via 1kΩ resistor	Digit common 4

If common cathode: transistor emitter → GND, collector → digit common pin.

If common anode: transistor emitter → 5V, collector → digit common pin, and invert logic in code.

Full Wiring Summary

Power:

- Arduino 5V → 74HC595 Pin 16 (VCC) and transistor collectors (if PNP for CA display)
- Arduino GND → 74HC595 Pin 8 (GND), transistor emitters (if NPN for CC display), and display GND

Control Lines:

- Arduino Pin 4 → 74HC595 Pin 14 (DS)
- Arduino Pin 5 → 74HC595 Pin 12 (Latch, STCP)
- Arduino Pin 6 → 74HC595 Pin 11 (Clock, SHCP)
- Arduino Pin 8 → Digit 1 transistor base (via 1kΩ resistor)
- Arduino Pin 10 → Digit 2 transistor base (via 1kΩ resistor)
- Arduino Pin 11 → Digit 3 transistor base (via 1kΩ resistor)
- Arduino Pin 12 → Digit 4 transistor base (via 1kΩ resistor)

Segments:

Q0–Q7 → 220Ω resistor → display segment pins a–g + DP

Code Upload (15 min) – Provide students with the following Arduino code for initial testing: (Current code only powers the motor for the conveyor with IR detection, final stage conveyor can be continuously powered or coded to stop as well)

```
// === MOTOR CONTROL ===
const int ENA = 9;    // PWM for motor speed (already used)
const int IN1 = 2;    // Motor direction
const int IN2 = 3;    // Motor direction
const int IR_PIN = 7;  // IR sensor input
// === 74HC595 SHIFT REGISTER PINS ===
const int DATA_PIN = 4;   // DS pin on 74HC595
const int LATCH_PIN = 5;  // STCP pin on 74HC595
const int CLOCK_PIN = 6;  // SHCP pin on 74HC595
// === DIGIT SELECT PINS (NPN transistors for common cathode) ===
const int DIGIT1 = 8;   // Leftmost digit
const int DIGIT2 = 10;
const int DIGIT3 = 11;
const int DIGIT4 = 12;
// === TIMING SETTINGS ===
const int motorSpeed = 250;          // PWM speed value
const unsigned long stopTime = 1000;    // Motor stop duration (ms)
const unsigned long ignoreTime = 2500; // Time IR sensor is ignored after detection (ms)
// === STATE TRACKING ===
unsigned long lastDetectionTime = 0;
bool ignoringSensor = false;
unsigned int objectCount = 0;
// === 7-seg digit patterns (common cathode, segments A–G + DP) ===
// Bit order: DP G F E D C B A
```

```
const byte SEGMENT_MAP[10] = {
  0b00111111, // 0
  0b00000110, // 1
  0b01011011, // 2
  0b01001111, // 3
  0b01100110, // 4
  0b01101101, // 5
  0b01111101, // 6
 0b00000111, // 7
 0b01111111, // 8
  0b01101111  // 9
};
const byte SEG_DASH = 0b01000000; // "-"
// === FUNCTION PROTOTYPES ===
void runMotorForward(int speed);
void stopMotor();
void displayNumber(int num);
void displayDash();
void sendToShiftRegister(byte data);

void setup() {
  pinMode(ENA, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IR_PIN, INPUT);
 pinMode(DATA_PIN, OUTPUT);
  pinMode(LATCH_PIN, OUTPUT);
  pinMode(CLOCK_PIN, OUTPUT);

  pinMode(DIGIT1, OUTPUT);
```

```
  pinMode(DIGIT2, OUTPUT);
  pinMode(DIGIT3, OUTPUT);
  pinMode(DIGIT4, OUTPUT);
  runMotorForward(motorSpeed);
}
void loop() {
  unsigned long currentTime = millis();
  // Resume IR sensor after ignore time
  if (ignoringSensor && (currentTime - lastDetectionTime >= ignoreTime)) {
    ignoringSensor = false;
                   }
// Handle IR detection if sensor is active
  if (!ignoringSensor) {
    bool objectDetected = (digitalRead(IR_PIN) == LOW);
  if (objectDetected) {
      objectCount++;
  stopMotor();
      delay(stopTime);
      runMotorForward(motorSpeed);
      ignoringSensor = true;
      lastDetectionTime = currentTime;
    }
  }

                   // Update display
                   if (ignoringSensor) {
                     displayDash(); // Show ---- during ignore period
                   } else {
                     displayNumber(objectCount);
                   }
```

```
            }
// === MOTOR FUNCTIONS ===
void runMotorForward(int speed) {
 digitalWrite(IN1, HIGH);
 digitalWrite(IN2, LOW);
 analogWrite(ENA, speed);
}
void stopMotor() {
 analogWrite(ENA, 0);
}
// === DISPLAY FUNCTIONS ===
void displayNumber(int num) {
 int thousands = (num / 1000) % 10;
 int hundreds  = (num / 100) % 10;
 int tens      = (num / 10) % 10;
 int ones      = num % 10;
 // Multiplexing - quickly light each digit
 digitalWrite(DIGIT1, LOW);
 sendToShiftRegister(SEGMENT_MAP[thousands]);
 digitalWrite(DIGIT1, HIGH);

 digitalWrite(DIGIT2, LOW);
 sendToShiftRegister(SEGMENT_MAP[hundreds]);
 digitalWrite(DIGIT2, HIGH);

 digitalWrite(DIGIT3, LOW);
 sendToShiftRegister(SEGMENT_MAP[tens]);
 digitalWrite(DIGIT3, HIGH);
 digitalWrite(DIGIT4, LOW);
 sendToShiftRegister(SEGMENT_MAP[ones]);
```

```
  digitalWrite(DIGIT4, HIGH);
}
void displayDash() {
  for (int i = 0; i < 4; i++) {
    digitalWrite(DIGIT1 + i, LOW);
    sendToShiftRegister(SEG_DASH);
    digitalWrite(DIGIT1 + i, HIGH);
  }
}
void sendToShiftRegister(byte data) {
  digitalWrite(LATCH_PIN, LOW);
  shiftOut(DATA_PIN, CLOCK_PIN, MSBFIRST, data);
  digitalWrite(LATCH_PIN, HIGH);
}
```

Testing Phase (as time allows) – Students upload code and verify motor & display operation.

Homework: Think about possible improvements for motor speed, conveyor spacing, and object detection.

Day 3 — Testing, Optimization, & Presentation

Objective: Students will finalize their builds, troubleshoot, and explain the real-world application of their conveyor system.

Activities:

1. **Warm-Up (5 min)** – Quick recap of yesterday's build steps.
2. **Complete Assembly & Debugging (20 min)** – Students fix wiring issues, adjust motor speeds, and ensure sensors detect objects correctly.
3. **Performance Testing (15 min)** – Groups run test batches of objects to verify:
    - Motors stop when the IR sensor detects an object.
    - The display counts objects correctly.
    - The conveyor resumes after 1000ms and ignores IR input for 2.5 seconds.
4. **Mini-Presentations (10 min)** – Each group explains:

- Their conveyor's role in a pharmaceutical plant.
- Any modifications made.
- How their system could be scaled for real production.

5. **Wrap-Up & Assessment Instructions (5 min)** – Explain summative quiz for next class.

Summative Assessment — 10 Questions

Multiple Choice / True-False (6)

1. Which type of conveyor is best suited for transporting pill bottles in a cleanroom?
   a) Belt conveyor
   b) Roller conveyor
   c) Pneumatic conveyor
   d) Chain conveyor
2. True/False: In pharmaceutical manufacturing, conveyors must be easy to clean to prevent contamination.
3. Which Arduino component stores data for the 7-segment display?
   a) L298N
   b) 74HC595
   c) IR Sensor
   d) DC Motor
4. The IR sensor in this project is used to:
   a) Stop the motor when an object is detected
   b) Measure motor speed
   c) Display numbers on the screen
   d) Count conveyor rotations
5. The motor driver pins ENA, IN1, IN2 control:
   a) LED lights
   b) Conveyor direction & speed
   c) Sensor detection
   d) Shift register
6. True/False: The Arduino code in this project makes the IR sensor stop reading for 2.5 seconds after an object is detected.

   **Short Answer **

7. Describe one advantage of belt conveyors in pharmaceutical manufacturing.
8. Explain how the 74HC595 shift register reduces the number of Arduino pins needed for the display.
9. In your own words, describe the sequence of actions that occur after the IR sensor detects an object.
10. Suggest one improvement to your conveyor system for real-world application.