Course Outline – Microprocessors (40S)

Instructor: Mr.Pulver
Contact: Pulver.Joe@bsd.ca
Course Code: EXC40S
Credit: 1.0
Prerequisite: Basic understanding of DC circuits and components (Electronics 20S/30S recommended). Introductory programming experience is beneficial but not required.
Suggested Grade Level: Grade 12
Delivery Format: Classroom instruction, lab activities, and project-based learning

Course Description

This course introduces students to microprocessor and microcontroller systems through hands-on exploration. Students will learn how microprocessors interface with sensors, actuators, and user inputs to create functional systems. Programming will be done using the Arduino IDE, with emphasis on coding fundamentals, debugging, and applying control structures.

The course is project-based: students begin with guided builds from online modules, then progress toward designing original microprocessor applications that integrate hardware and software. Along the way, students will build skills in schematic interpretation, breadboarding, soldering, project documentation, and troubleshooting.

The course culminates in a capstone project, where students propose, design, and implement their own Arduino-based system, showcasing their ability to combine coding, electronics, and problem-solving.

Unit Breakdown

Unit 1: Introduction to Microprocessors & Arduino (1–2 weeks)

• What is a microprocessor/microcontroller?

• Arduino hardware and the role of I/O pins

• Introduction to the Arduino IDE (setup & loop functions)

• Uploading and running a simple sketch (Blink example)

• Safety and best practices in electronics labs

• Mini-Project: First Arduino circuit – blinking LED with modifications

Unit 2: Coding Fundamentals (2–3 weeks)

• Structure of Arduino sketches

• Variables, loops (for/while), and conditionals (if/else)

• Serial monitor for debugging

• Functions and modular code in Arduino C

• Mini-Project: Interactive LED patterns (students design custom blinking sequences)

Unit 3: Digital and Analog Inputs/Outputs (3 weeks)

• Digital inputs: buttons, switches

• Analog inputs: potentiometers, light sensors

• Digital outputs: LEDs, buzzers

• PWM outputs: LED dimming, motor speed control

• ADC/DAC concepts in Arduino

• Mini-Project: Light-sensitive night lamp (photoresistor + LED)

Unit 4: Sensors and Actuators (3 weeks)

• Using sensors: temperature, distance, tilt, etc.

• Controlling actuators: DC motors, servos, relays

• Combining multiple inputs and outputs in a project

• Mini-Project: Temperature-controlled fan (sensor + motor driver)

Unit 5: Project Development with CTC 101 Modules (4–5 weeks)

• Guided builds from Arduino CTC 101 kit:

• - Traffic light system

• - Servo sweep & positioning

• - Alarm system with sensors

• - Basic motor control applications

• Emphasis on troubleshooting, documentation, and teamwork

• Project Task: Modify an existing CTC 101 project to add a new feature

Unit 6: Communication & Advanced Applications (2–3 weeks)

• Introduction to communication protocols (UART, I2C, SPI – simplified through Arduino libraries)

• Serial data transfer and debugging

• Multi-device communication (e.g., Arduino + peripheral board)

• Mini-Project: Arduino serial monitor data logger (temperature readings)

Unit 7: Capstone Project (4 weeks)

• Proposal and planning of student-designed project

• Integration of sensors, actuators, and custom code

• Independent building and testing with teacher guidance

• Project report and final demonstration

• Examples: Automated greenhouse, robotic arm, home automation system, wearable sensor

Assessment Breakdown

Employability Skills	10%
Labs, Theory, and Skills	60%
Final Project (Capstone)	30%