

---

# **Privacy Guide for P2 Developers Using Claude Code**

Protecting Your Code While Leveraging AI Assistance

P2 Knowledge Base Project

September 2025

## Contents

<b>1</b>	<b>Privacy Guide for P2 Developers Using Claude Code</b>	<b>2</b>
1.1	Welcome to AI-Assisted P2 Development . . . . .	2
1.2	Understanding Claude Code in Your Development Workflow . . . . .	2
1.3	Your Code, Your Privacy . . . . .	2
1.4	Smart Strategies for AI-Assisted Development . . . . .	3
1.5	Data Security Best Practices . . . . .	4
1.6	Understanding AI Limitations and Risks . . . . .	5
1.7	Practical Scenarios and Solutions . . . . .	6
1.8	FAQ: Common Privacy Concerns . . . . .	7
1.9	Building Trust Through Transparency . . . . .	8
1.10	Quick Reference Card . . . . .	9
1.11	Getting Started Safely . . . . .	10
1.12	Conclusion . . . . .	10
1.13	13. Additional Resources . . . . .	11

# 1 Privacy Guide for P2 Developers Using Claude Code

## 1.1 Welcome to AI-Assisted P2 Development

As you begin your journey developing for the Propeller 2 microcontroller with Claude Code as your AI partner, understanding privacy and data security is essential. This guide explains how to protect your intellectual property, maintain code confidentiality, and use AI tools safely while developing embedded systems.

---

## 1.2 Understanding Claude Code in Your Development Workflow

### 1.2.1 What Claude Code Is

Claude Code is an AI-powered development assistant that helps you:

- Write and debug Spin2 and PASM2 code
- Understand P2 architecture and capabilities
- Solve complex embedded programming challenges
- Generate boilerplate code and templates
- Explain instruction behaviors and timing

### 1.2.2 What Claude Code Isn't

- **Not a data collector** - It doesn't harvest your code for training
  - **Not a telemetry system** - No usage tracking or analytics
  - **Not connected to other users** - Your sessions are isolated
  - **Not a permanent storage** - Conversations aren't archived indefinitely
- 

## 1.3 Your Code, Your Privacy

### 1.3.1 How Your Code Stays Yours

**Session Isolation** - Each Claude Code session is independent - Your code isn't visible to other developers - No cross-contamination between projects - Clean slate with each new conversation

**Training on Your Data - You Choose - Default:** Claude doesn't learn from your conversations - **Opt-in Available:** You can choose to help improve Claude by allowing training - **Your Control:** This is entirely

your decision - **Proprietary Protection:** Most developers keep this OFF for commercial work - **Open Source:** Consider opting IN when working on public projects

**Temporary Processing Only** - Code is processed only during active sessions - No long-term storage of your implementations - Conversations expire and are deleted - No permanent record of your development process

### 1.3.2 Best Practices for Code Privacy

#### 1. Sanitize Sensitive Information

```
1 ' Instead of this:
2 API_KEY = "sk-proj-abc123xyz789" ' Company API key
3
4 ' Do this:
5 API_KEY = "YOUR_API_KEY_HERE" ' Replace with actual key
```

#### 2. Abstract Proprietary Logic

```
1 ' Instead of sharing your secret sauce:
2 PRI calculateProprietaryFilter(input) : output
3 ' [200 lines of proprietary DSP code]
4
5 ' Share the interface and problem:
6 PRI calculateProprietaryFilter(input) : output
7 ' Need help optimizing this DSP routine
8 ' Input: 16-bit samples at 48kHz
9 ' Output: Filtered signal
10 ' Current execution: 850 cycles, need < 600
```

**3. Use Generic Examples** - Replace company-specific constants with generic values - Substitute real product names with descriptive placeholders - Remove customer-specific requirements from queries - Focus on technical challenges, not business logic

---

## 1.4 Smart Strategies for AI-Assisted Development

### 1.4.1 Protecting Intellectual Property

**Compartmentalize Your Queries** - Break complex systems into generic components - Ask about patterns, not implementations - Focus on public algorithms and techniques - Keep system architecture discussions high-level

**Example: Safe vs. Risky Queries**

☒ **Safe Query:** “How can I implement I2C communication on P2 Smart Pins?”

☒ **Risky Query:** “Here’s my complete motor control system for our new drone product...”

☒ **Safe Query:** “What’s the most efficient way to implement a FIR filter in PASM2?”

☒ **Risky Query:** “This is our proprietary sensor fusion algorithm that gives us market advantage...”

### 1.4.2 Working with Confidential Projects

**The Air-Gap Approach** 1. Develop core proprietary logic offline 2. Use Claude Code for generic helper functions 3. Integrate components in your private environment 4. Test complete system without AI assistance

**The Abstraction Method** 1. Create interfaces and stubs for sensitive components 2. Get AI help with non-critical surrounding code 3. Implement secret sauce privately 4. Maintain clear boundaries between public and private code

---

## 1.5 Data Security Best Practices

### 1.5.1 Before You Start Coding

**Environment Setup** - [ ] Review your company’s AI usage policies - [ ] Understand what code can be shared - [ ] Set up local backups of all work - [ ] Create clear project boundaries - [ ] Document what’s shareable vs. confidential

**Project Planning** - [ ] Identify proprietary components early - [ ] Plan which parts need AI assistance - [ ] Prepare sanitized test cases - [ ] Create generic examples for testing - [ ] Establish code review procedures

### 1.5.2 During Development

#### Safe Sharing Practices

```
1 ' ⓧ GOOD: Generic timing question
2 "How do I ensure this PASM2 loop executes in exactly 16 cycles?"
3
4 ' ⓧ BAD: Revealing proprietary timing
5 "Our product requires 16.384MHz sampling for patent #123456"
6
7 ' ⓧ GOOD: Abstract the problem
8 "I need to read 8 sensors sequentially within 100µs"
9
10 ' ⓧ BAD: Expose the application
11 "Our medical device monitors these specific biomarkers..."
```

**Code Review Checklist** Before sharing code with Claude Code: - [ ] Remove all API keys and credentials - [ ] Strip out customer or company names - [ ] Delete proprietary comments - [ ] Replace magic numbers with generic values - [ ] Verify no NDA-covered information remains

### 1.5.3 After Development Sessions

**Session Hygiene** 1. Export important code to your local system 2. Clear conversation history when switching projects 3. Don't rely on Claude Code for code storage 4. Maintain your own version control 5. Document learnings in your private notes

---

## 1.6 Understanding AI Limitations and Risks

### 1.6.1 Technical Limitations

**Claude Code Cannot:** - Access your local file system directly - See other conversations or sessions - Remember previous projects after clearing - Share your code with other users - Train itself on your implementations (unless you opt in) - Use your code for training without permission

**You Are Responsible For:** - Verifying all generated code - Testing on actual P2 hardware - Maintaining security of your systems - Protecting proprietary information - Complying with company policies

### 1.6.2 Privacy Considerations

**What Anthropic Can See** - Aggregated usage statistics (not your code) - System performance metrics - Error reports and crashes - General feature usage patterns

**What Anthropic Cannot Use (Default Settings)** - Your conversations for training (unless you opt in)  
- Your specific implementations to train future models - Your code to improve Claude (without your permission)

**What Remains Private (Even with Opt-in)** - Other users never see your code - Your sessions remain isolated - Your competitive advantage stays yours - Direct customer data should never be shared

---

## 1.7 Practical Scenarios and Solutions

### 1.7.1 Scenario 1: Debug Helper Pattern

**Need:** Help debugging a complex Smart Pin configuration

**Safe Approach:**

```
1 ' Share the problematic configuration
2 wrpin(P_SYNC_TX | P_OE, pin)
3 wxpin($1_0000 | (8-1), pin)
4 wypin(0, pin)
5 ' Ask: "Why isn't this UART transmitting?"
```

**What NOT to Do:**

```
1 ' Don't share the entire proprietary communication protocol
2 ' Don't reveal what device you're interfacing with
3 ' Don't include customer-specific baud rates
```

### 1.7.2 Scenario 2: Algorithm Optimization

**Need:** Optimize a control loop for speed

**Safe Approach:** - Share the mathematical operation needed - Provide generic timing constraints - Ask about PASM2 optimization techniques - Focus on instruction-level improvements

**Protected Elements:** - Keep control parameters private - Don't reveal system dynamics - Hide application-specific constants - Protect tuning values

### 1.7.3 Scenario 3: Learning New Features

**Need:** Understand CORDIC engine operations

**Safe Approach:**

```
1 ' Ask about generic CORDIC operations
2 "How do I use QROTATE for coordinate transformation?"
3 "What's the precision of QSIN/QCOS functions?"
4 "Can you show a generic FFT using CORDIC?"
```

**Maintain Privacy:** - Don't reveal signal processing applications - Keep frequency ranges generic - Hide specific accuracy requirements - Protect domain-specific uses

---

## 1.8 FAQ: Common Privacy Concerns

### 1.8.1 Q: Can other developers see my code?

**A:** No. Each Claude Code session is completely isolated. Your code is never visible to other users or their sessions.

### 1.8.2 Q: Does Claude learn from my bugs and solutions?

**A:** By default, no. Claude doesn't train on your conversations unless you explicitly opt in. You can choose to allow training to help improve Claude, but this is entirely optional and most developers keep it disabled for proprietary work.

### 1.8.3 Q: What happens to my code after a conversation?

**A:** Conversations are retained temporarily for technical purposes (like conversation continuity). By default, they are not used for training and are eventually deleted. If you've opted in to training, Anthropic may use conversations to improve Claude, but this is your choice.

### 1.8.4 Q: Can I use Claude Code for commercial products?

**A:** Yes, but follow your company's AI usage policies and protect proprietary information using the practices in this guide.

### 1.8.5 Q: Should I share my entire project with Claude Code?

**A:** No. Share only the portions where you need assistance. Keep proprietary logic and sensitive information local.



### 1.8.6 Q: What if I accidentally share confidential information?

**A:** Start a new conversation immediately. Don't continue in the same session. Review what was shared and inform your security team if required by policy.

### 1.8.7 Q: Can Anthropic see my P2 product designs?

**A:** Anthropic has technical safeguards but may process conversations for safety and quality. If you've opted in to training, your conversations may be used to improve Claude. For maximum protection of trade secrets, keep training OFF and avoid sharing highly confidential designs regardless of settings.

### 1.8.8 Q: Is it safe to use Claude Code for hobbyist projects?

**A:** Yes! Hobbyist and open-source projects are ideal for AI assistance since there are fewer confidentiality concerns.

---

## 1.9 Building Trust Through Transparency

### 1.9.1 Our Recommendations

1. **Start Small** - Test with non-critical code first
2. **Build Confidence** - Learn what works safely
3. **Establish Patterns** - Develop your own privacy practices
4. **Stay Informed** - Keep up with AI privacy developments
5. **Share Experiences** - Help the P2 community learn

### 1.9.2 Community Resources

- **P2 Forums** - Share sanitized examples and learnings
  - **GitHub** - Contribute to open-source P2 projects
  - **Documentation** - Help improve public P2 resources
  - **Tutorials** - Create generic learning materials
-

## 1.10 Quick Reference Card

### 1.10.1 ☒ SAFE TO SHARE

- Generic P2 programming questions
- Public algorithm implementations
- Standard peripheral interfaces
- Common debugging scenarios
- Learning queries about P2 architecture
- Open-source code examples
- Published techniques and patterns

### 1.10.2 ☒ NEVER SHARE

- API keys or credentials
- Customer names or data
- Proprietary algorithms
- Patent-pending techniques
- Company trade secrets
- NDA-covered information
- Competitive advantages

### 1.10.3 ☒ SANITIZE FIRST

- Product-specific code
  - Timing-critical implementations
  - Custom protocols
  - Performance optimizations
  - System architectures
  - Business logic
  - Hardware interfaces
-

## 1.11 Getting Started Safely

### 1.11.1 Your First Week with Claude Code

**Day 1-2: Learn the Basics** - Practice with P2 tutorials - Ask about instruction syntax - Explore Smart Pin configurations - No proprietary code yet

**Day 3-4: Build Confidence** - Try debugging assistance - Request code reviews - Learn optimization techniques - Use only example code

**Day 5-7: Establish Workflow** - Identify shareable components - Practice sanitization techniques - Develop your privacy habits - Create template queries

### 1.11.2 Long-term Success

**Monthly Review:** - Audit your sharing practices - Update sanitization techniques - Review any close calls - Refine your workflow

**Continuous Improvement:** - Learn from the community - Share safe examples - Contribute to documentation - Build collective knowledge

---

## 1.12 Conclusion

Using Claude Code for P2 development can significantly accelerate your embedded programming journey while maintaining complete privacy and security. By following these guidelines, you can:

- Leverage AI assistance effectively
- Protect your intellectual property
- Maintain competitive advantages
- Contribute to the P2 community
- Build amazing embedded systems

Remember: AI is a powerful tool, but you remain in control. Use it wisely, protect what matters, and enjoy the enhanced productivity it brings to your P2 development.

---

**Happy Coding!**

*Stay safe, stay smart, and build something amazing with P2 and Claude Code.*

*Document Version: 1.0*

*Last Updated: September 2025*

*For the latest version, check the P2 Knowledge Base*

**1.13 13. Additional Resources**

- Anthropic's Privacy Policy
- P2 Documentation
- P2 Community Forums
- Claude Code Documentation

*This guide is provided for educational purposes. Always consult your organization's policies and legal requirements regarding AI tool usage.*