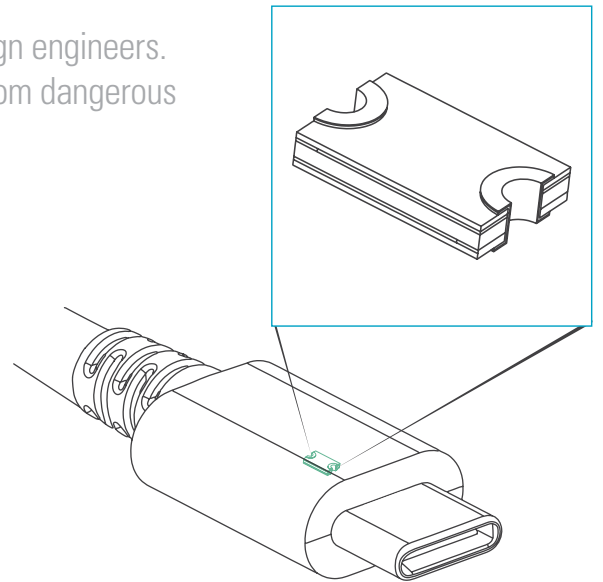
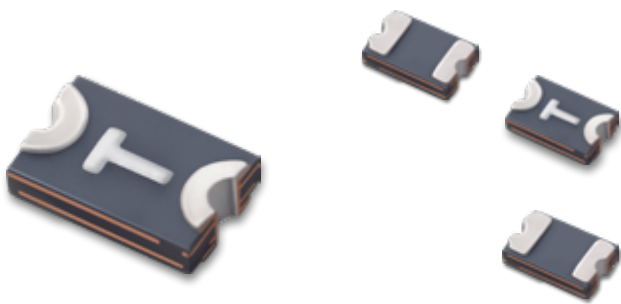


Over Heating **Protection**  
Fully **USB-PD** Compliant



Implementing USB Type-C charging presents challenges to design engineers. Littelfuse temperature indicator solutions help protect cables from dangerous overheating due to resistive faults from power line to ground.



Fast-Charging  
for  
Mobile



Reversible  
Connector



Fast  
Data  
Transfer



Low  
Temperature  
with **setP™**

## setP™ Key Characteristics

Function	Applications	Ordering Number	Indicating Temperature	Resistance @ 25°C	Indicating Resistance	Footprint
Over-temperature Protection	Captive cable USB Type-C Chargers	SETP0805-100-SE	100°C ±10°	12Ω or less	35kΩ or greater	0805
Over-temperature Protection	USB Type-C to Type-C Cables	SETP0805-100-CC	100°C ±10°	6Ω or less	35kΩ or greater	0805

## Keeps the Plug Surface Cool

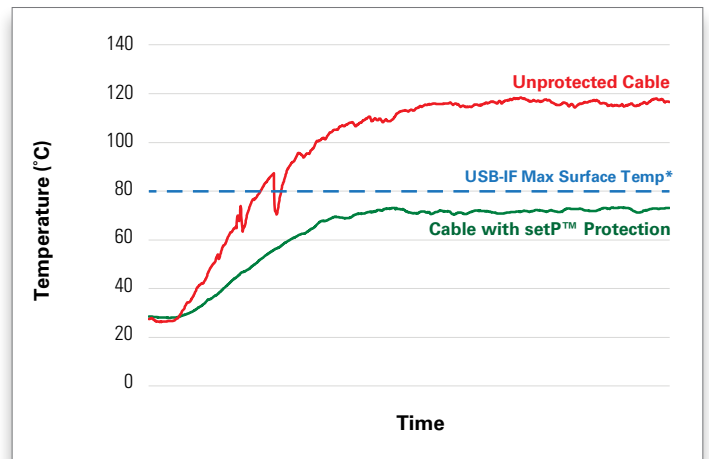
### Problem Condition

- More Power
  - Smaller pin-to-pin
  - Universal Fit
- Higher Risk

Easier for contamination or deformed pins to cause a fault. Higher power increases risk of thermal event.

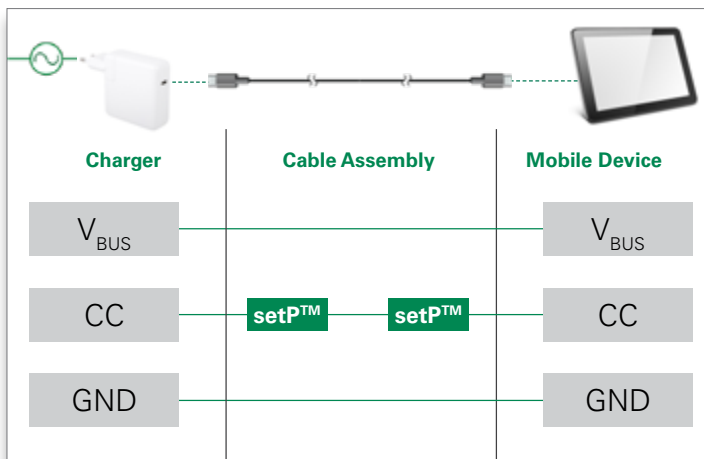


## Surface Temperature During Over-temperature Fault



\* Reference temperature set by USB-IF within Table 6-14 of the USB Type-C Cable and Connector Specification.

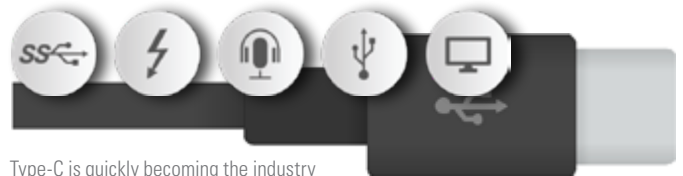
## Circuit Diagram & Protection Explanation



setP™, located inside the Type-C plug, senses the temperature of the USB Type-C Connector.

- Charger is connected to the AC power line and cable is connected to the mobile device
- Fault occurs causing heat (either at charger or mobile device side),
  1. setP™ senses heat, then resistance ( $R_{setP}$ ) increase
  2.  $R_{setP}$  increase causes voltage on CC Line to increase beyond specified value\*
  3. System assumes cable detached due to voltage on CC being higher than specified value\*, thus  $V_{BUS}$  power is turned off
- The system is protected
- To clear the fault: Disconnect the cable and remove debris

\* vOpen value is defined by USB-IF as either 1.65V or 2.75V



Type-C is quickly becoming the industry standard connector